모니터링/점검

Tibero Database Shared Memory와 Semaphore 확인

TMOTI054

Website
www.tmaxtibero.com

안내서 정보
안내서 제목: Tibero Database Shared Memory 와 Semaphore 확인
발행일: 2025-11-25
소프트웨어 버전: 6FS05, 6FS06, 6FS07, 6FS07PS, 7FS01, 7FS02, 7FS02PS
안내서 버전: 1.1

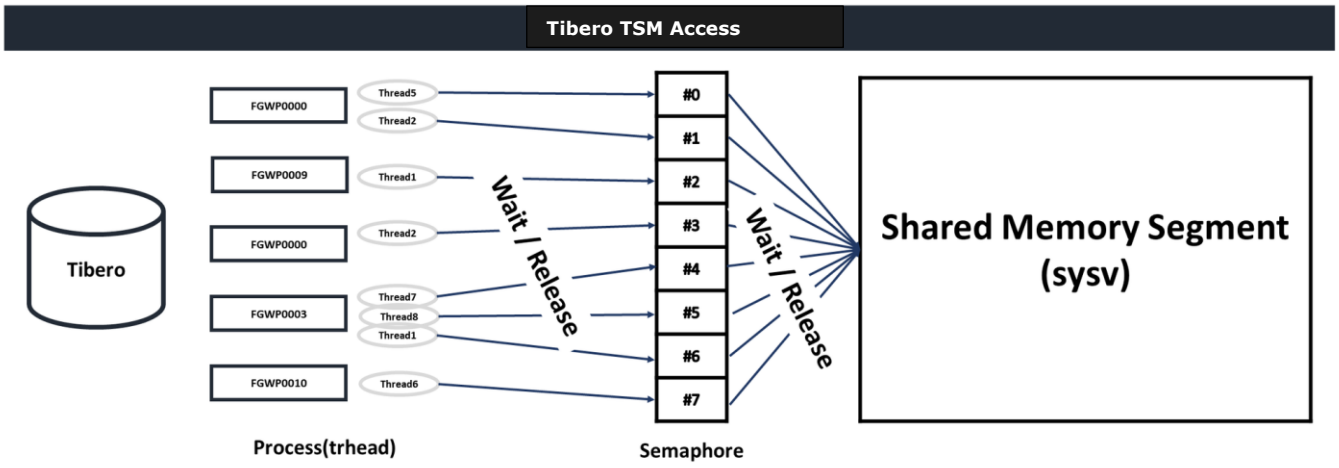
제, 개정 이력

안내서 버전	개정일자	개정 사유 및 내용	비고
1.0	2023.10	최초 제정	작성자:윤준수
1.1	2025.11.25	Kernel Parameter overview 추가	작성자:고다혜

1. Overview

본 문서에서는 ***어떤 Process가 "Shared Memory"와 "Semaphore"의 주소***를 사용하고 있는지 알아보는 방법에 대해 기술하였습니다.

Database에서 여러 Process들이 공유해야 하는 메모리 영역을 사용할 때 ***mmap, sysv, posix***와 같은 방식을 사용할 수 있습니다. ***Tibero Database(이하: Tibero)의 경우 sysv 방식을 사용***하고 있습니다. Tibero에서는 여러 Process들간 공유하는 메모리 공간인 ***TSM 영역을 "Shared Memory"*** 영역에 생성하며 Process들간 ***TSM 영역 간섭을 방지하기 위해 "Semaphore"***을 통해 "Shared Memory"에 접근하고 있습니다.



※ **리눅스 환경에서 이루어진 테스트입니다. UNIX 환경에서도 비슷하게 적용할 수 있지만 내용이 다를 수 있으니 UNIX 환경에서는 참고만 부탁드립니다.**

※ **기술된 내용을 인용하여 쉘 스크립트 작성하시길 추천 드립니다.**

2. Kernel Parameter

공유 메모리(Shared Memory)와 세마포어(Semaphore)는 Tibero가 여러 프로세스를 동시에 실행할 때 서로 충돌 없이 메모리를 공유하고, 작업 순서를 조절하기 위해 사용하는 기능입니다.
이 기능들은 리눅스에서 제공하는 프로세스 간 통신(IPC) 방식 중 하나로, Tibero가 정상적으로 작동하기 위해 꼭 필요합니다.
아래 커널 파라미터들은 이러한 메모리 공유와 동기화에 직접적인 영향을 주며, 값이 적절하게 설정되지 않으면 Tibero 기동 실패, 비정상 종료 후 메모리 잔류, Semaphore 부족 등의 문제가 발생할 수 있습니다.

2.1 커널 파라미터 확인 명령어

Shared Memory 관련 파라미터 확인

```
$ sysctl -a | grep -E 'kernel\.shm' | grep -v -E 'shm_next_id|shm_rmid_forced'
kernel.shmall = 10000000
kernel.shmmax = 3221225472
kernel.shmmni = 4096
```

Semaphore 관련 파라미터 확인

```
$ ipcs -l | grep –A4 Semaphore
------ Semaphore Limits --------
max number of arrays = 10000
max semaphores per array = 10000
max semaphores system wide = 32000
max ops per semop call = 10000
```

2.2 kernel.sem

kernel.sem은 프로세스 간 동기화를 위한 Semaphore 설정으로, 총 4개의 값으로 구성됩니다.

• 경로: /proc/sys/kernel/sem

• 구성 요소:

- semmsl : 세마포어 세트 당 최대 세마포어 개수

- semmns : 시스템 전체 세마포어 최대 개수이며 일반적으로 semmsl × semmni 이상으로 설정해야 합니다.

- semopm : 호출(call) 단위로 가능한 최대 세마포어 연산 수

- semmni : 세마포어 세트 최대 개수

최소 설정값

• semmsl: 2 / 권장값 : (Tibero 전체 Thread 수 × 2)

• semmns: (Tibero 전체 Thread 수 × 2)

• semopm: 최소값 2(=SEMMSL) / 권장값 = (Tibero 전체 Thread 수 × 2)

• semmni: Tibero 전체 Thread 수

참고
Tibero 전체 Thread 수는 대략 아래와 같습니다.
MAX_SESSION_COUNT + (MAX_SESSION_COUNT/WTHR_PER_PROC) + 500

※ kernel.sem 값은 최대치를 높게 설정해도 문제가 없으므로 여유롭게 설정하는 것을 권장합니다.

2.3 kernel.shmall

특정 시점에 시스템에서 사용 가능한 공유 메모리 총량을 의미합니다.

2.4 kernel.shmmax

하나의 공유 메모리 세그먼트가 가질 수 있는 최대 크기를 의미하며, 단위는 Byte입니다.

Tibero의 TSM 메모리 크기는 shmmax보다 같거나 작아야 하며,

shmmax가 충분히 크지 않으면 기동 시 Shared Memory 생성 오류가 발생할 수 있습니다.

2.5 kernel.shmmni

공유 메모리 세그먼트의 최대 개수를 의미하며, 리눅스 기본값은 4096입니다.

3. Problem

Tibero 비정상 기동과 다운 그리고 기타 이유로 인해 ***"Shared Memory" 및 "Semaphore" 영역이 정리되지 않은 경우*** 두 공간 모두 Tibero만을 위해 사용된다면 다행이지만 만약 ***다른 Process들이 생성한 영역이 존재***한다면 다음과 같은 문제를 겪을 수 있습니다.

1) **Tibero를 root에서 기동하고 Tibero Process가 Kill된 경우**
   A. "Shared Memory", "Semaphore" 영역 모두 정리되지 않고 root로 남아 있음
   B. TiberoProcess가 아닌 기존에 root로 기동 된 영역이 있다면 문제 소지 있음
2) **Tibero만 사용하는 줄 알고 ipcrm -a 명령을 root에서 수행한 경우**
   A. root는 OS 최고 권한 계정이라 하위 User들의 모든 공간 초기화 수행
   B. 타 솔루션 및 데이터베이스에 영향을 받음
3) **Tibero 기동 또는 다운 중 Ctrl-C 또는 터미널 창이 끊기면서 도중 취소된 경우**
   A. "1)"과 동일하게 비정상 기동/다운이 될 수 있음
   B. "Shared Memory", "Semaphore"가 정리되지 않음
   C. 어떤 메모리 주소 값을 제거해야 하는지 확인 어려움

EX) Oracle ASM(파일시스템)을 사용하는 경우 ASM grid 유저에서 "Shared Memory" 영역을 사용하고 있어 ***ipcrm -a 수행하는 경우 Oracle ASM의 공간을 초기화하여 파일시스템 마운트가 해제***되는 경우가 발생할 수 있습니다.

Oracle ASM 운영 환경

```
$ ipcs -m
------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 65536      grid     640        16384      2
0x00000000 98305      grid     640        16384      2
0x00000000 131074     grid     640        16384      2
0x00000000 163843     grid     640        16384      2
0x30ca87e7 41         tibero       640        2147483648 12
```

EX) ***기타 Process들이 "Shared Memory" 영역을 사용***하는 경우 다른 Process한테 영향을 끼칠 수 있습니다. 만약 단일 서버에 여러 개의 데이터베이스가 운영 중에 있고 "Shared Memory", "Semaphore" 영역을 사용하고 있는 경우 ***ipcrm -a 수행 시 예기치 못한 상황***이 발생할 수 있습니다.

단일 서버에 기타 Process 또는 타 DBMS 운영 중인 경우

```
$ ipcs -m
------ Shared Memory Segments --------
key        shmid      owner            perms      bytes      nattch     status
0x30ca87e7 41         tibero_out        640        2147483648 12
0x0dc036e3 43         tibero_in         640        2147483648 14
0x98d6bb70 45         postgres          640        2147483648 14
```

"Shared Memory", "Semaphore" 영역을 초기화할 때 OS 재기동 또는 ipcrm(-a, -m, -s) 명령을 통해 초기화 합니다. 이러한 경우 ***Tibero Process에서 사용했던 ID를 알 수 없는 경우*** 출처를 알지 못하는 값들까지 지워지게 될 수 있어 예기치 못한 상황이 발생할 수 있습니다. ***"Shared Memory", "Semaphore" 영역의 출처를 알고 초기화하는 것이 안전합니다.***

※ tbdown clean_shm 명령을 실행하면 Tibero가 생성한 "Shared Memory"는 정리되지만, "Semaphore" 영역은 제거되지 않습니다.

또한 Shared Memory를 다른 프로세스가 계속 사용하고 있는 상태라면, 해당 프로세스가 갑자기 메모리 없이 동작하게 되어 비정상 종료가 발생할 수 있기 때문에 Shared Memory를 삭제하지 않고 그대로 남겨둡니다.

4. Shared Memory 확인 방법

"Shared Memory" ID 확인

```
$ ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x30ca87e7 41         root       640        2147483648 12
0x0dc036e3 43         root       640        2147483648 14
0x98d6bb70 47         root       640        2147483648 14
```

OS에 생성된 ***"Shared Memory" 영역을 확인***합니다. shmid 정보만 알면 됩니다.

Tibero Process 리스트 확인

```
$ cat $TB_HOME/instance/$TB_SID/.proc.list
Tibero 7   start at (2023-10-05 07:55:53) by 0
shared memory: 140135677796352 size: 2147483648
shm_key: -1730757776 1 sem_key: 1579222643 205 listener_pid: 300656 listener_port:
28629 listener_special_port: 28630 epa_pid: -1
300655 MONP
300657 MGWP
300658 FGWP0000
300659 FGWP0001
300660 PEWP0000
300661 PEWP0001
300662 PEWP0002
300663 PEWP0003
300664 PEWP0004
… 생략 …
```

Process가 기동 되어 있는 경우라면 쉽게 Process ID를 확인할 수 있지만 비정상 기동 되었거나 ***다운 된 경우 .proc.list 파일의 내용 또는 slog를 통해 Process ID***를 확인할 수 있습니다.

Tibero Process의 lsof를 통해 확인하기

```
$ lsof -p 300655 |grep SYS
tbsvr    300655 root   DEL     REG                  0,1                    47
/SYSV98d6bb70
```

lsof 명령어 사용이 가능하다면 ***Tibero Process ID를 -p 옵션***으로 주어 확인합니다. "Shared Memory" 영역을 만들기 위해서는 ***System V IPC(sysv) 객체를 통해 생성되기 때문에 grep SYS***로 찾을 수 있습니다. 위의 ***명령 결과 47이라는 "Shared Memory" ID를 확인***할 수 있습니다. 다만 Tibero Process가 ***다운되어 있는 상태라면 이 방법으로는 확인할 수 없습니다.***

Tibero Process의 maps 파일을 통해 확인

```
$ cat /proc/300655/maps |grep SYS |grep delete
7f73e14ba000-7f74614ba000 rw-s 00000000 00:01 47    /SYSV98d6bb70 (deleted)
```

OS에서 만들어낸 Process 메모리 맵(요약) 파일을 사용합니다. /proc/TiberoPID/maps 파일에서 ***System V IPC(sysv) 객체 정보를 확인***합니다. ***명령 결과 47이라는 "Shared Memory" ID를 확인***할 수 있습니다. 다만 ***Tibero Process가 다운되어 있는 상태라면 이 방법으로는 확인할 수 없습니다.***

Tibero Process의 smaps 파일을 통해 확인

```
$ cat /proc/300655/smaps |grep SYS |grep delete
7f73e14ba000-7f74614ba000 rw-s 00000000 00:01 47    /SYSV98d6bb70 (deleted)
```

OS에서 만들어낸 Process 메모리 맵(상세) 파일을 사용합니다. /proc/TiberoPID/smaps 파일에서 ***System V IPC(sysv) 객체 정보를 확인***합니다. ***명령 결과 47이라는 "Shared Memory" ID를 확인***할 수 있습니다. 다만 ***Tibero Process가 다운되어 있는 상태라면 이 방법으로는 확인할 수 없습니다.***

OS의 shm 파일을 통해 확인

```
$ cat $TB_HOME/instance/$TB_SID/.proc.list
Tibero 7   start at (2023-10-05 07:55:53) by 0
shared memory: 140135677796352 size: 2147483648
shm_key: -1730757776
```

… 생략 …

```
$ cat /proc/sysvipc/shm
       key       shmid perms                size  cpid  lpid nattch   uid   gid  cuid  cgid      atime      dtime      ctime                 rss
swap
 818579431          41 640          2147483648 298989 298989     12     0     0     0     0 1696487924 1696487924 1696487923          1224269824
0
 230700771          43 640          2147483648 299433 300212     14     0     0     0     0 1696492206 1696492206 1696488602          1234423808
0
-1730757776         47 640          2147483648 300613 300613     14     0     0     0     0 1696492555 1696492556 1696492553          1232125952
0
```

Tibero Process가 다운된 경우에는 $TB_HOME/instance/$TB_SID/.proc.list 파일을 통해 ***다운 전 Tibero Process ID를 확인***할 수 있습니다. 다만 Process가 살아있지 않아 lsof, maps, smaps와 같은 OS에서 만들어낸 ***Tibero Process 정보가 정리되어 확인할 수 없습니다.*** 이러한 경우 ***OS에서는 아직 Tibero Process가 비정상 다운 되기전의 "Shared Memory" 영역이 제거되지 않아 /proc/sysvipc/shm의 정보를 사용하면 확인***할 수 있습니다. .proc.list 파일의 shm_key와 shm 파일의 key의 값이 일치하는 shmid을 알 수 있습니다.

5. Semaphore 확인 방법

"Semaphore" ID 확인

```
[root@tibero /]$ ipcs -s

------ Semaphore Arrays --------
key        semid      owner      perms      nsems
0x1f174b6e 1015808    root       600        2
0x1f174b6f 1015809    root       600        2
0x1f174b70 1015810    root       600        2
0x1f174b71 1015811    root       600        2
0x1f174b72 1015812    root       600        2
0x1f174b73 1015813    root       600        2
0x1f174b74 1015814    root       600        2
0x1f174b75 1015815    root       600        2
```

"Semaphore" 결과의 ***semid 정보를 통해 확인***할 수 있습니다.

Tibero Process 리스트 확인

```
$ cat $TB_HOME/instance/$TB_SID/.proc.list
Tibero 7   start at (2023-10-05 07:55:53) by 0
shared memory: 140135677796352 size: 2147483648
shm_key: -1730757776 1 sem_key: 1579222643 205 listener_pid: 300656 listener_port:
28629 listener_special_port: 28630 epa_pid: -1
300655 MONP
300657 MGWP
300658 FGWP0000
300659 FGWP0001
300660 PEWP0000
300661 PEWP0001
300662 PEWP0002
300663 PEWP0003
300664 PEWP0004
… 생략 …

$ ipcs -s -i 7

Semaphore Array semid=1147338
uid=0   gid=0   cuid=0 cgid=0
mode=0600, access_perms=0600
nsems = 2
otime = Thu Oct  5 07:55:56 2023
ctime = Thu Oct  5 07:55:55 2023
semnum     value      ncount     zcount     pid
0          0          0          0          300661
1          0          1          0          300661
```

"Semaphore"의 영역의 경우 "Shared Memory" 영역만큼 확인해볼 수 있는 방법이 많지 않습니

다. ipcs -s -i SemaphoreID 명령을 통해 최근에 해당 "Semaphore"에 접근한 Process ID를 확인
할 수 있어 위와 같은 방법으로 어떤 Process가 사용한 Semaphore인지 확인할 수 있습니다.