Real Alternative DBMS ALTIBASE, Since 1999

UNIX Memory Management

2010. 03

Document Control

Change Record

Date	Author	Change Reference
2010-03-11	lim272	Created

Reviews

Date	Name (Position)

Distribution

Name	Location

목차

개요

본 문서에서는 유닉스(UNIX) 시스템에서 프로세스가 요청하는 메모리에 대해 어떻게 관리되는지 개략적인 이해를 위해 만들어 졌다. 따라서, 심도 있는 이해는 각 벤더에서 제공하는 문서 등을 참고하도록 한다.

Solaris 시스템의 관리 기법

솔라리스 운영체제의 메모리 관리에 대해 설명한다.

메모리 할당

솔라리스는 reserved라는 형태로 메모리를 할당한다. reserved영역은 swap영역에 존재한다. 즉, 어떤 프로세스가 10M를 요청하면 swap영역에 10M를 먼저 할당해 놓고 실제 해당 프로세스가 메모리에 접근할 때 물리적 메모리에 10M를 할당하는 형태로 동작한다.

따라서, 솔라리스는 Swap영역이 부족하면 어떠한 프로세스도 동작할 수 없다.

```
-bash-4.0$ /usr/sbin/swap -s
total: 3610640k bytes allocated + 1939792k reserved = 5550432k used, 36742272k available
```

위의 박스에 예처럼 swap정보로 볼 때 실제 물리적인 메모리가 여유 있을 때라도 Swap영역이 쓰여 지고 있는걸 확인할 수 있는데 이는 솔라리스 자체가 메모리 요청에 대해 무조건 reserved영역을 먼저 할당하는 정책을 갖고 있기 때문이다.

아래는 코드상으로 VSZ/swap영역의 변화를 보여 준다.

	Memory Allocation	Reserved(Swap)	VSZ
1. 요청	P = malloc(100M)	100M	0M
2. 실제 사용	For (i=0; i<10M;i++) *(p+i) = 1	90M	10M

실제 코드상으로 alloc을 하여도 메모리가 즉시 증가하지 않는다. 이후 실제 접근 시점에 메모리 사용량이 증가하는 것을 확인할 수 있다.

메모리 부족

솔라리스는 기본적으로 여유 있는 메모리는 파일캐쉬로 사용한다. 이 조건은 기본적으로 물리적 메모리가 lotsfree로 설정된 값(전체 메모리의 1/64) 이상으로 여유 있을 때에만 파일캐쉬로 사용한다. (5.7 이하에서는 파일캐쉬를 free메모리가 필요로 할 때 우선적으로 선택될 수 있도록 옵션을 줘야 했으나 5.8 이상부터는 기본적으로 파일캐쉬 역시 free메모리에 잡힌다. 따라서, AIX/HP와 달리 사용자가 별도의 설정을 할 필요는 없다.)

그런데, 만일, lotsfree이하로 유지가 되기 시작하면 시스템은 메모리 페이지들을 검색하여 최근 사용되지 않는 페이지들을 찾아내기 시작한다. (vmstat 정보에서 sr 부분에 이러한 검색 수치가 나온다.)

lotsfree를 유지하기 위해 이러한 자주 접근되지 않는 페이지들을 메모리에 내리고 lotsfree수준을 채우도록 동작한다. 이와 같은 동작을 swap(swapping)이 발생한다고 지칭한다. (vmstat상에서는 fr은 메모리의 freePage된 개수를 의미하는데 free되었다는 말은 해당 메모리상의 페이지가 변경된 정보가 있다면 디스크로 갱신되게 됨으로 이와

같은 swapping이 발생하는 상태에서는 디스크I/O가 빈번하게 발생함으로 시스템 전체의 성능이 저하되는 현상을 보인다.)

pmap

솔라리스는 다음과 같이 프로세스의 실 사용 부분에 대해 상세하게 조회할 수 있는 유틸을 제공하고 있다.

```
# pmap -F 22748
0000000100D22000       8560K rwx--    /home1/hjkim/altibase/5.1.5.72/bin/altibase
000000010157E000     488320K rwx--      [ heap ]
FFFFFFFF72EFE000          8K rw--R      [ stack tid=74 ]
FFFFFFFF730FC000         16K rw--R      [ anon ]
FFFFFFFF732FC000         16K rw--R      [ anon ]
FFFFFFFF73C00000      10240K rw-s-   dev:118,46 ino:45717892
```

맨 위는 프로세스 메모리가 될 것이고 heap영역이 메모리DB등이 위치하는 영역이 된다. anon의 의미는 MMAP_PRIVATE 맵핑을 가진 페이지에 대해 초기 접근 시 영역을 의미한다. ino등이 있는 것은 mmap으로 올라온 리두 로그 버퍼 영역이라고 보면 된다.

AIX 시스템의 메모리 관리 정책

AIX 운영체제의 메모리 관리에 대해 설명한다.

메모리의 분류

AIX의 메모리 사용에 대해 이해를 위해 먼저 메모리의 분류에 대한 정의를 설명한다.

분류	설명
Persistent	JFS의 파일캐쉬로 사용되는 영역
Client	CDROM, NFS, JFS2 의 파일캐쉬로 사용되는 영역
Computational	Process stack, heap, shared Memory 등의 영역

이해를 위해 svmon으로 나온 결과를 가지고 설명해 보도록 한다. (svmon의 결과는 특별한 표기가 없는 한 모두 page단위이며 1page는 기본적으로 4K이다.)

```
Shell> svmon -G
               size       inuse        free         pin     virtual
memory      2031616     1779678      251938      474697     1682009
pg space    4128768      495129

               work        pers        clnt       other
pin          404863           0           0       69834
in use      1225427           5      554246
```

위의 결과를 아래 표에서 먼저 설명한다.

항목 (붉은 박스)	설명
size	전체 물리적 메모리의 페이지 개수를 의미한다. 실제 1page는 4,096byte임으로 7936M의 메모리를 갖는 시스템임.
inuse	(Computational + Persistent)의 실제 사용 중인 물리적인 메모리의 페이지 개수
free	물리적 메모리에서 사용 중이지 않은 페이지 개수
pin	Swap out할 수 없는 물리적 메모리의 페이지 개수
virtual	VMM (Virtual Memory Manager)에 의해 생성된 페이지 개수
pg space	Paging space 공간의 사용량

붉은 박스 바깥쪽의 pin의 합계가 붉은 박스 내의 pin의 용량과 같고 inuse 역시 동일하다. 여기서 알아 둘 것은 동일한 inuse로 사용 중이어도 일부분은 파일캐쉬로 사용 중이라는 사실을 확인할 수 있다는 점이다.

메모리의 할당

SUN과 다르게 미리 reserved영역을 할당하지 않고 deferred형태로 동작한다. 즉, 동작 중에 실제로 메모리가 부족할 경우에만 swap영역을 사용하는 형태로 동작한다고 이해하면 된다.

메모리의 부족

AIX는 기본적으로 여유 메모리를 모두 파일캐쉬로 사용하려고 노력한다. 따라서 메모리가 부족할 경우에는 다음과 같이 동작한다. 만일, 파일캐쉬로 사용 중인 메모리의 양이 MAXPERM이상이면 무조건 파일캐쉬에서 메모리를 steal한다. MAXPERM과 MINPERM사이이면 파일캐쉬와 computational 메모리 중에 I/O가 적을 것이라고 판단되는 쪽에서 steal을 하게 된다. 따라서, MAXPERM의 설정값 크기에 따라서는 프로세스 입장에서는 잘 사용되던 메모리의 일부가 paging space 영역으로 swap되면서 재 접근 시에 디스크I/O를 유발하여 성능상 지터(jitter)현상이 발생하는 경우가 있을 수 있다.

ALTIBASE를 사용할 경우 이러한 지터 현상을 가능한 배제하기 위해 AIX5.2ML04 이상에서는 몇 가지 프로퍼티를 설정하도록 권고하고 있다. 자세한 설정 부분은 『ALTIBASE 환경 설정 가이드 For AIX』 문서나 AIX에서 배포하는 성능 튜닝과 관련된 기술문서를 참고하도록 한다.

관련 항목	설명
MAXPERM	물리적 메모리가 파일캐쉬로 사용되는 점유율의 최대치 (soft limit)
MINPERM	물리적 메모리가 파일캐쉬로 사용되는 점유율의 최소치
NUMPERM	실제 파일캐쉬로 사용되는 영역의 점유율 (vmtune, vmo등으로 확인)
MAXCLIENT	NFS, JFS2 등의 파일캐쉬로 사용되는 점유율의 최대치
stric_maxperm	1 로 설정할 경우 MAXPERM을 유지하게 함.
lru_file_repage	0 으로 설정 시 메모리 부족 시에 발생하는 steal에 대해 JFS2 등의 파일캐쉬에서만 발생하도록 강제로 지정.

svmon

AIX에서는 svmon을 통해 프로세스의 실제 메모리 사용량을 자세히 확인할 수 있다.

```
Shell> svmon -P 356528
------------------------------------------------------------------------------
Pid     Command          Inuse       Pin      Pgsp   Virtual 64-bit Mthrd   16MB
356528  altibase         37992      8380     59579     96475      Y      Y      N

PageSize        Inuse         Pin        Pgsp     Virtual
s    4 KB       29224        8332       59579       87707
m   64 KB         548           3           0         548
```

Svmon은 기본적으로 스냅샷을 제공하는 정보가 아니며 통계성 정보로 보아야 한다. 따라서, (ps v [process id]) 와 같은 명령어로 확인하는 메모리 사용량과 svmon의 결과는 다를 수 있다. 다만, svmon에서는 pgsp항목을 주목할 필요가 있다. 이 부분이 증가하고 있다는 점은 실제로 메모리가 부족하거나 또는 computation memory영역이 steal되어 swapping되었다는 의미이고 ALTIBASE 입장에서는 성능 저하를 유발할 수 있는 문제임으로 확인 후 파일캐쉬 설정을 조정하도록 해야 한다.

ps 명령을 통해 아래와 같이 실제 메모리의 사용량을 알아낼 수도 있다.

Shell> ps v 356528

PID	TTY	STAT	TIME	PGIN	**SIZE**	RSS	LIM	TSIZ	TRS	%CPU	%MEM
356528		- A	29:37	4700	**290328**	77688	xx	17933	3396	0.1	1.0

ps결과와 svmon의 결과에서 분석상의 문제는 svmon의 inuse부분의 합계가 실제 ps결과의 SIZE와 일치해야 하지만 page-out된 경우에는 실제로 ps쪽이 더 크게 표시된다.

HP 시스템의 메모리 관리 정책

HP 운영체제의 메모리 관리에 대해 설명한다.

메모리의 할당

HP의 메모리 할당 정책은 arena라고 부르는 부분과 관련이 깊다. 자세한 사항은 man-Page를 통해 malloc으로 확인하도록 하며 요점만 설명하면 시스템은 메모리를 할당하기 위해 메모리풀(arena)을 관리하는데 쓰레드 프로그램의 경우 이 arena를 통해 메모리를 할당 받게 된다. 만일, 쓰레드의 개수가 많이 증가할 경우 arena의 개수를 조정함으로써 동시성 측면에서 성능 향상을 도모할 수 있다. (non-thread의 경우는 오직 1 개의 arena를 통해 메모리를 할당 받게 된다.)

Shell> **export _M_ARENA_OPT=16:8**

위와 같이 환경을 설정하면 16 개의 arena를 통해 쓰레드들이 메모리를 할당 받게 되는데 만일 arena가 가진 메모리풀이 부족 해지면 (8*4096byte)단위로 메모리풀을 확장하는 형태로 동작하겠다는 의미이다. (expansion의 단위가 너무 클 경우 급격하게 메모리가 증가하는 현상이 발생할 수 있음으로 이 환경 변수를 설정할 때에는 많은 테스트를 필요로 한다. 기본값은 8:32 이다.)

메모리의 부족

Swapping정책은 여타 운영체제와 다르지 않아 별도의 설명은 생략한다.

메모리 사용량 확인

공통적으로 Glance를 통해 확인하는 방법이 가능하다.

Shell> glance 를 구동한 후 “m”으로 분기하면 전체 시스템 메모리 상태를 확인할 수 있으며 “s”를 누른 후 프로세스를 지정하여 “M”을 누르면 프로세스의 메모리 사용 형태를 확인할 수 있다.

pmap이 있는 버전의 경우 pmap을 사용할 수 있다.

rx6600:[/] **pmap 12379**

12379: /altibase_home/bin/altibase -p boot from admin

OFFSET	VSZ	RSZ	TYPE	PRM FILE
0	4K	4K	SD(170)	r-- [nullderef]
4000000000000000	15.1M	11.5M	SC(2)	r-x [text]
6000000000000000	901M	658M	PD	rw- [data]
9fffffff7d67f000	72K	64K	PD	rw- [uarea]
9fffffff7d7af000	72K	64K	PD	rw- [uarea]

HP에서 프로세스의 정확한 메모리 사용은 pmap명령으로 확인할 수 있다. 솔라리스와 비슷한 결과 형태를 보여 주고 있다.

HP도 AIX와 동일하게 파일캐쉬에 대한 설정을 조정할 수 있다. 이 값들은 전체적인 성능과도 연관이 있기 때문에 상황에 따라 설정 권고치가 가변적이지만 일반적으로 5%(min)/15%(max)를 설정 권고한다.

커널 항목	설명
dbc_max_pct	파일캐쉬로 사용할 메모리의 최대 임계치
dbc_min_pct	파일캐쉬로 사용할 메모리의 최소 임계치

Linux 시스템의 메모리 관리 정책

Linux 운영체제의 메모리 관리에 대해 설명한다.

Linux 메모리 관리

리눅스는 파일캐쉬 부분에 대해서는 AIX와 유사한 메모리 사용 정책을 갖는다. 즉, 여유 메모리는 모두 파일캐쉬로 사용하려고 시도한다. 단, 커널 2.6 부터는 이 부분에 대한 제한을 두게 되는데 다음으로 파일캐쉬의 사용량을 제한 시킬 수 있다.

```
Shell> cat /proc/sys/vm/swappiness

설정 시에는 sysctl을 사용한다.
```

기본적으로 60(%)으로 설정되어 있으며 조금 복잡한 산술식이 존재하지만 간단하게 swappiness에 설정된 값 이상으로 물리적 메모리가 사용되기 시작하면 무조건 swapping을 하기 시작한다. 이것은 리눅스 시스템 자체가 설정값 이하를 파일캐쉬로 확보하기 위한 노력으로 swapping을 발생 시키는 것이다. 하지만, 이 부분은 사용자가 원하던 원치 않던 swapping으로 인한 비용을 유발함으로써 파일캐쉬를 통한 성능상의 이점 보다 오히려 더 나쁜 시스템 성능을 가져올 수 있기 때문에 비록 현재, ALTIBASE에서는 이에 대해 특별한 권고를 하고 있지 않지만 MySQL등에서는 이 커널 값에 대해 “0”으로 설정할 것을 권고하고 있음을 참고하도록 한다.

Linux 메모리 확인

Top 또는 pmap 명령으로 프로세스의 메모리 사용량을 확인할 수 있다.

Misc.

기타 공통적이거나 혹은 별도의 숙지할 사항을 설명한다.

메모리의 기본 크기

각 운영체제 별로 메모리 관리를 최소 단위인 페이지 단위로 수행하며 개별 크기는 아래와 같다.

운영체제	SUN	AIX	HP
크기 (단위:byte)	8192	4096	4096

왜 vsz은 줄지 않는가?

일반적으로 운영체제는 프로세스가 사용한 메모리 영역을 프로세스가 종료될 때에만 free영역으로 반납한다. 즉, 프로세스에서 사용자가 명시적으로 할당 받은 메모리 영역에 대해 free()를 호출하여도 그 즉시, 해당 영역을 해제하지 않는다. 이는 해당 프로세스가 해제한 메모리 영역을 다시 재사용할 것이라고 기대하는 부분과 운영체제의 메모리 관리자가 프로세스에 의해 해제된 영역이 프라그먼트 형태로 남아 이를 다시 재구성하여 할당 가능한 세그먼트의 free-list로 만들어야 할 경우 커널 비용이 성능에 큰 영향을 줄 수 있기 때문이다.

따라서, 운영체제는 프로세스가 free()를 하더라도 실제 모니터링 툴 등을 통해 VSZ의 크기가 줄지 않는 현상을 보게 되는데 이는 앞서 설명한 이유 때문이다.

알티베이스㈜
서울특별시 구로구 구로 3 동 182-13
대륭포스트 2 차 1008 호
02-2082-1000
http://www.altibase.com

대전사무소
대전광역시 서구 둔산동 921
주은리더스텔 901 호
042-489-0330

기술지원본부
서울특별시 구로구 구로 3 동 182-13
대륭포스트 2 차 908 호
02-2082-1000

솔루션센터
02-2082-1114
http://support.altibase.com