

Real Alternative DBMS ALTIBASE, Since 1999
ALTIBASE 운영을 위한 HPUX 설정 가이드

2010. 03.18



Copyright © 2000~2013 ALTIBASE Corporation. All Rights Reserved.

Document Control

Change Record

Date	Author	Change Reference
2010-03-19	durusari	Created

Reviews

Date	Name (Department)
2010-03-19	lim272(SP), hankimchi (TS), chk0618(TS), khhan (SC)

Distribution

Name	Location

목차

개요	4
커널파라미터.....	5
공유메모리.....	5
세마포어.....	6
파일캐시.....	6
리소스 제한.....	7
변경방법.....	7
사용자 환경설정	9
리소스 제한.....	9
환경변수.....	10
터미널 환경.....	11
멀티쓰레드 관련 패치.....	11
요약	13
설정예제.....	13

개요

HPUX(Hewlett Packard UniX) 운영체제(Operating System)에서 ALTIBASE를 설치하고 운영하기 위한 커널파라미터(kernel parameter)의 적절한 설정 값 및 각종 사용자 환경 설정에 대한 가이드를 제시한다.

본 문서에서는 ALTIBASE가 설치되기 전 설정해야 할 운영체제관련 사항에 대해서만 가이드를 제시하며 ALTIBASE 자체 설정을 위한 ALTIBASE 프로퍼티(property) 설정파일(alibase.properties)에 관해서는 별도문서 『ALTIBASE 설정파일 가이드』를 참조하도록 한다.

커널파라미터

HPUX 운영체제에서 ALTIBASE를 운영할 경우 변경해야 할 커널파라미터의 종류와 어떤 이유로 설정을 변경해야 하는지에 대해 설명한다. 마지막 부분에는 HPUX에서 커널파라미터를 변경하는 방법에 대해 소개한다.

각 커널파라미터와 관련된 상세한 사항은 HP에서 제공하는 가이드를 참조하도록 한다.

공유메모리

응용프로그램을 개발하는 경우 두 개 이상의 프로세스가 정보를 주고 받아야 할 필요성이 발생한다. 운영체제는 이를 위해 IPC (Inter Process Communication)라고 부르는 자원을 제공하고 있다. 여러 IPC자원 중에 두 개 이상의 프로세스가 정보를 주고 받기 위해 사용되는 메모리영역을 공유메모리(shared memory)라고 부른다.

공유메모리는 사용자가 단위를 지정하여 한 개 이상의 영역으로 나누어 설정할 수 있다. 한 개의 공유메모리 영역을 세그먼트(segment)라고 부른다.

예를 들어 사용자는 10 MB로 1 개의 세그먼트를 갖는 공유메모리를 설정할 수도 있고 10 MB영역을 10 개의 세그먼트로 구성하여 100 MB의 공유메모리를 설정할 수도 있다.

ALTIBASE 는 기본설정으로 메모리DB를 프로세스의 힙(heap)영역에 적재하지만 사용자의 설정에 따라서는 공유메모리에 적재하여 사용할 수 있다. (이와 관련된 설정방법은 『ALTIBASE 설정파일 가이드』 문서를 참고하도록 한다)

혹은, DBMS와 접속하는 클라이언트의 통신버퍼를 공유메모리를 통해 주고 받을 수 있도록 설정할 수 있다. 따라서, 메모리를 사용하는 부분에 대한 최대크기나 세그먼트의 개수제한에 대한 커널파라미터 설정을 올바르게 변경해야 한다.

일반적인 유닉스 시스템에서 공유메모리와 관련된 커널파라미터는 아래와 같다.

커널파라미터	설명	권장 값
shmmax	한 개의 공유메모리 세그먼트의 최대크기	2G+1
shmmni	시스템내의 공유메모리 세그먼트의 최대개수	500 이상
shmseg	한 개의 공유메모리에 접근 가능한 프로세스의 최대개수	200 이상

메모리DB를 공유메모리에 적재할 경우 고려사항

메모리DB를 공유메모리에 적재할 경우에는 아래 사항들을 추가로 고려해야 한다.

- shmmax
shmmax의 경우는 구동단계에서는 ALTIBASE 설정파일 내에 STARTUP_SHM_CHUNK_SIZE 항목의 크기만큼 한 개의 세그먼트가 만들어지기 때문에 STARTUP_SHM_CHUNK_SIZE 보다 큰 값으로 설정하도록 한다.
- shmmni
메모리DB의 사용량이 증가하게 되면 ALTIBASE는 자동적으로 메모리영역을 증가시킨다. 이 경우 역시 세그먼트가 새롭게 생성되는데 이 세그먼트의 크기는 ALTIBASE설정파일 내에 EXPAND_CHUNK_PAGE_COUNT * 32K 단위로 세그먼트가 생성되기 때문에 shmmni는 가능한 여유 있게 설정하도록 한다.

세마포어

공유메모리와 동일하게 IPC자원의 일종이다. 차이점은 공유메모리가 데이터를 프로세스 간에 공유하기 위한 방법으로 제공되는 반면에 세마포어는 프로세스 간의 동기화를 구현하기 위한 방법으로 운영체제에서 제공되는 자원이다. 동기화란 특정자원이나 객체에 대해 한 시점에 하나의 프로세스만이 접근하여 변경 가능하도록 제어 하는 것을 의미한다. (뮤텍스(mutex)와 다르게 조회는 동시에 가능하게 하다.)

ALTIBASE는 앞에서 설명한대로 로컬서버 내에 ALTIBASE 서버와 클라이언트가 존재할 경우 양단간의 통신버퍼로 공유메모리를 사용하는데 이때, 메모리에 대한 동시성을 제어해야 할 필요가 있다. (동시에 쓰기를 진행하면 안되기 때문이다.)

이 경우 메모리에 대한 읽기/쓰기에 대한 접근권한의 제어를 위해 세마포어 오퍼레이션(semaphore operation)이 발생하게 된다.

세마포어 오퍼레이션에 따라 프로세스는 대기상태가 되거나 혹은 진행상태가 될 수 있다. 이러한 세마포어 오퍼레이션은 동시다발적으로 발생되기 때문에 세마포어의 개수 및 오퍼레이션에 대한 적합한 커널파라미터 설정이 필요하다.

일반적인 유닉스 시스템에서 세마포어와 관련된 커널파라미터는 아래와 같다.

커널파라미터	설명	권장 값
semmns	시스템 내 세마포어의 최대개수이며 16 바이트의 커널메모리가 할당된다.	8192
semmni	세마포어 세트의 최대개수로 65535 이내에 설정가능하며 하나의 세트당 84 바이트의 커널메모리가 할당된다.	5029
semmsl	한 개의 세마포어 세트에 존재하는 세마포어의 최대개수이며 논리적으로 semmns와 같거나 작아야 한다. 너무 크게 잡으면 몇 개의 세마포어 ID가 시스템 전체의 세마포어를 독점할 수 있다.	2000
semmap	semget 호출에 의해 얻어지는 세마포어 공간을 의미	5024
semmnu	시스템 내의 undo structure의 최대개수	1024
semume	한 개의 프로세스당 undo entry의 최대개수	512
semopm	semop 시스템 콜에서 처리하는 operation의 최대개수	512
semvmx	한 개의 세마포어의 최대값을 제한하며 32767 보다 크게 지정하지 않는다.	32767

파일캐시

본 커널파라미터는 변경이 필수적인 요소는 아니나, 적절한 파일캐시(file cache) 설정으로 ALTIBASE가 사용하는 메모리 영역에 대한 스왑아웃(swap out) 발생요건을 억제, 스왑핑(swapping)으로 인한 운영체제 계층의 Disk I/O 지연시간이 ALTIBASE의 성능저하로 이어지는 현상을 최소화하기 위해 권고하는 사항이다.

파일캐시란 주기억장치와 보조기억장치간의 속도차이로 인한 병목을 해소하기 위해 운영체제차원에서 관리되는 일종의 시스템버퍼이다. 이러한 파일캐시는 운영체제 저마다의 고유한 정책(policy)에 의하여 관리되나 공통적으로 스왑정책과 직접적인 상관관계를 가진다.

스왑핑이란 자체는 주기억장치보다 더 큰 응용프로그램이나 데이터 파일을 다룰 수 있게 하는 유용함을 가지고 있으나, DBMS와 같이 장기상주 형의 응용프로그램이 운영되는 시스템에서는 스왑핑으로 인한 운영체제 계층의 Disk I/O 지연시간으로 인해 DBMS의 응답시간이 불규칙해지거나 최악의 경우에는 행(hang)과 같은 현상이 발생할 수 있기에 시스템 용도에 따라 고려가 요소가 된다.

따라서, ALTIBASE의 일관적인 응답시간 보장을 위해서는 스왑이 최대한 발생하지 않도록 관련 파일캐시 및 스왑 커널들을 사전에 설정하는 것이 바람직하다.

HPUX에서의 설정

아래 커널파라미터를 통해 파일캐시를 조정하는 것을 권장한다.

커널파라미터	설명	권장 값
dbc_min_pct	파일캐시의 최소 크기를 전체 메모리 대비 백분율로 제한, 기본값은 5%로 11.31 부터는 filecache_min 로 명칭이 변경됨.	5%
dbc_max_pct	파일캐시의 최대 크기를 전체 메모리 대비 백분율로 제한 기본값은 50%로 11.31 부터는 filecache_max 로 명칭이 변경됨.	5~20%

참고로 dbc_max_pct에 대한 HP에서의 권고안은 8G 이하 시스템에서는 20%, 8G 이상 시스템에서는 10%를 권고하므로 이를 참고하도록 한다.

리소스 제한

HPUX 경우, 리소스 제한 항목 중 일부는 일반적으로 흔히 사용하는 ulimit을 사용한 사용자 환경설정파일 설정이 아닌 아래와 같은 커널파라미터 변경을 통하여 설정한다.

커널파라미터	설명	권장 값
maxdsiz	하나의 32bit 프로세스가 할당 가능한 데이터 세그먼트 합계	2G
maxdsiz_64bit	하나의 64bit 프로세스가 할당 가능한 데이터 세그먼트 합계	1T / 최대 4T (예측되는 ALTIBASE 프로세스 크기 산정)
max_thread_proc	하나의 프로세스가 가질 수 있는 최대 쓰레드 개수	600 이상
maxfiles	하나의 프로세스가 동시에 오픈(open)할 수 있는 최대 파일 개수	2048 이상
nproc	사용자당 생성 가능한 프로세스의 개수	6142
maxusers	nproc, ncallout, ninode, nfile의 기본값의 근간	124

“maxusers”의 경우, HPUX11.23 부터는 사라졌으므로 HPUX11.23 이상에서는 고려하지 않아도 된다. HPUX11.23 미만에서는 “nproc” 를 직접적으로 설정하지 않고 “maxusers”를 124 로 설정하는 것으로 대체한다.

변경방법

HPUX 경우, 커널파라미터 수정을 위해서 Tunable 커널파라미터 유틸리티(HPUX 11.11 에서는 kmtune, HPUX 11.23 부터는 kctune을 주로 사용)나 시스템의 모든 커널파라미터를 변경 가능한 sam이라는 유틸리티를 주로 사용한다. 여기서는 kctune을 사용한 예를 다룬다.

기본적으로 root 계정으로의 접속을 필요로 하며 실시간으로 적용되는 커널파라미터라 할지라도 올바르게 적용하기 위해서 변경 후 시스템을 재 구동을 하는 것을 권장한다.

공유메모리

이전에 별도의 설정이 없었다면 공유메모리 관련 커널파라미터 변경으로 shmmni, shmseg만 변경해도 무방하다.

```
# kctune shmmni=500
# kctune shmseg=200
```

세마포어

이전에 별도의 설정이 없었다면 세마포어 관련 커널파라미터 변경으로 semmns, semmni, semmnu, semume 만 변경해도 무방하다.

```
# kctune semmns=8192
# kctune semmni=5029
# kctune semmnu=5029
# kctune semume=5029
```

파일캐시

dbc_max_pct의 경우, HP의 권고치의 따라 8G 이하 시스템에서는 20%, 8G 이상 시스템에서는 10%를 권고한다.

```
# kctune dbc_min_pct=5
# kctune dbc_max_pct=10
```

리소스 제한

위에서 언급한 리소스 제한 관련 커널파라미터를 변경하는 방법은 아래와 같다.

```
# kctune maxdsiz=2147483648
# kctune maxdsiz_64bit=4396972765184
# kctune max_thread_proc=5029
# kctune maxfiles=5029
# kctune nproc=6142
```

HPUX11.23 미만에서는 "nproc" 를 직접적으로 설정하지 않고 "maxusers"를 124 로 설정하는 것으로 대체한다.

사용자 환경설정

HPUX 운영체제에서 ALTIBASE 운영을 위한 시스템 내의 사용자계정의 리소스 제한, 환경변수 및 각종 환경설정에 대해 설명한다.

설정과 관련된 명령어 및 구체적인 사항은 HP에서 제공하는 가이드를 참조하도록 하는 것으로 대체한다.

리소스 제한

유닉스 운영체제에서는 사용자계정 단위로 사용 가능한 리소스에 대한 논리적인 제한 설정한다. 이 리소스 제한 항목 중, 안정적인 서비스 운영을 위해 확장시켜야 하는 일반적인 유닉스 운영체제에서의 항목들은 아래와 같다.

항목	설명	권장 값
data seg size(data)	하나의 프로세스 데이터영역의 최대크기	unlimited
file size (fsize)	생성될 수 있는 파일의 최대크기	unlimited
open files (nofiles)	하나의 프로세스가 동시에 접근 가능한 파일의 최대개수	unlimited
max memory size (rss)	사용 가능한 메모리의 최대크기	unlimited
virtual memory (memory)	사용 가능한 가상메모리의 최대크기	unlimited
max user process	사용자당 생성 가능한 프로세스의 개수	unlimited

어디까지나 특정 사용자가 사용하는 메모리 및 데이터파일 영역 확장 시 물리적인 리소스가 풍부하더라도 논리적인 제한으로 인해 발생할 수 있는 문제를 사전에 제거하기 위한 것으로, 이 설정으로 인해 다른 프로세스에게 끼치는 영향은 없기에 해당 운영체제에서 허용하는 최대값(가능하다면 unlimited)으로 설정하는 것을 권고한다.

예를 들어, open files의 의미는 프로세스가 접근하는 파일뿐만 아니라 통신소켓의 개수까지 포함하고 있다. 만일, 이 값이 10으로 제한된 환경에서 ALTIBASE 를 운영할 경우 동시에 접속 가능한 세션은 10 개 이상은 불가능하다는 의미이다. (실제로 ALTIBASE가 사용하는 파일까지 고려하면 접속 가능한 세션이 없을 수도 있다.)

변경방법은 ulimit 명령을 사용한 환경설정파일 편집 및 시스템 리소스 설정파일 편집 또는 해당 운영체제 별로 제공하는 커널관련 유틸리티 활용이 있다.

Hard-Limit & Soft-Limit

리소스 제한 값은 hard-limit, soft-limit 이라는 개념으로 구분된다. hard-limit은 시스템어드민계정(root)외에는 변경할 수 없는 커널차원의 리소스 제한 최대값을 의미하며 soft-limit은 현재 사용자계정 내에서는 hard-limit까지 변경이 가능하다는 것을 의미한다. (상세한 내용은 ulimit -S / -H 옵션을 참고한다.)

soft-limit은 사용자가 접속을 하여 세션을 유지하는 동안 유효하며 변경을 하면 즉시 반영된다. 하지만 동일 사용자계정의 다른 세션이 연결되면 기존의 soft-limit이 반영되기 때문에 편의상 ulimit 명령어를 사용자계정의 환경설정파일에 추가하곤 한다.

하지만, 이러한 방법은 전역적인 hard-limit에 의해 의도한 설정이 되지 않을 수 있으므로 환경설정파일을 이용한 사용자계정 단위 적용보다는 시스템 차원의 리소스 설정파일 편집을 통하여 체계적으로 적용하는 것을 권고한다.

단, HPUX는 시스템 차원의 리소스 설정파일이 없으므로 리소스 제한을 위한 hard-limit 변경 시 sam이나 kctune을 사용한다.

환경변수

필수적인 환경변수는 아래와 같다. 아래 환경변수를 설정하고자 하는 사용자의 셸(shell)의 문법(syntax)에 따라 사용자계정의 환경설정과일에 반드시 설정하도록 한다.

환경변수	설명
ALTIBASE_HOME	ALTIBASE를 설치한 경로를 지정한다.
PATH	ALTIBASE_HOME/bin 에 있는 바이너리 및 셸 스크립트를 경로에 상관없이 활용하기 위해 경로를 지정하여 준다.
LD_LIBRARY_PATH	ALTIBASE의 동적 라이브러리를 사용할 경우 라이브러리가 존재하는 경로를 지정하여 준다.

이와 더불어 HPUX에만 존재하는 아래 환경변수도 필수적으로 추가하여야 한다.

환경변수	설명
SHLIB_PATH	동적 라이브러리를 32 비트로 링크할 경우에 설정하도록 한다.

멀티쓰레드 응용프로그램 위한 설정(1)

멀티쓰레드(multi-thread)기반의 응용프로그램인 ALTIBASE를 위해 별도 환경변수 설정이 필요하다. 참고로 본 문서에서는 대표적인 것만 언급을 하나, HPUX에서 지원하는 멀티쓰레드 관련 환경변수는 모두 고려할 필요가 있음을 유의해야 한다.

아래 요소는 모두 필수적인 사항이다. 그 중에서도 MxN 쓰레드모델관련 환경변수인 PTHREAD_FORCE_SCOPE_SYSTEM는 반드시 설정하도록 한다.

환경변수	설명	비고
PTHREAD_FORCE_SCOPE_SYSTEM	쓰레드 경합(contention) 영역을 시스템으로 설정	HPUX 11.23 부터 지원
PERF_ENABLE	유저 영역 sleep queue 연산 일부를 생략	HPUX 11.23 만 지원
PTHREAD_FAST_SHARED_OBJECTS	Shared Object에 대한 Private 알고리즘 적용	HPUX 11.31 부터 지원
PTHREAD_DISABLE_HANDOFF	다중 CPU환경에서 적용	HPUX 11.23 부터 지원

PERF_ENABLE의 경우, HPUX 11.31 부터는 PTHREAD_FORCE_SCOPE_SYSTEM에 포함되므로 HPUX 11.31 이상에서는 고려하지 않아도 된다.

멀티쓰레드 응용프로그램 위한 설정(2)

이 역시 멀티쓰레드기반의 응용프로그램을 위한 설정으로 정해진 권장 값을 제시할 수 없는 튜닝적인 요소이나 초기에 고려되어야 하는 사항이기에 언급한다.

특정 프로세스에 대한 메모리 요구를 위해 하나의 메모리할당영역만 사용하는 운영체제에서는 멀티쓰레드 응용프로그램의 여러 쓰레드가 동시에 메모리 요청(malloc, free)을 하였을 때 잠금 연산으로 인한 경합이 발생하므로 성능저하가 발생한다.

이러한 문제를 개선하기 위해 HPUX는 아레나(arena)라 불리는 메모리할당영역을 프로세스당 최대 64 개까지 제공, 아레나 간에 쓰레드를 분산하여 메모리 요청으로 인해 발생하는 잠금 경합을 감소시킬 수 있다.

아래 환경변수로 설정이 가능하다.

환경변수	설명	비고
_M_ARENA_OPTS=x:y	x: 프로세스당 할당될 아레나 개수로 기본값은 8 이다. [범위: 1-64] y: 아레나의 확장 단위로 메모리 페이지 개수로 나타내며 기본값은 32 이다. [범위: 1-4096]	아레나 개수(x)가 클수록 시스템 메모리 사용량이 늘어나고 작을수록 ALTIBASE의 성능이 저하되므로 적절한 수치로 설정 필요.

기본적으로 멀티쓰레드 응용프로그램이 아닌 경우는 설정여부와 관계없이 한 개의 아레나로 작동하며 유효하지 않은 값으로 설정할 경우, 기본값으로 동작한다.

예를 들어, 다음과 같이 설정한다면 멀티쓰레드 응용프로그램에 한하여 24 개의 아레나를 할당하고 각 아레나는 확장 시 64*4KB(일반적인 메모리 페이지 크기) 단위로 증가하게 되는 것이다.

```
# export _M_ARENA_OPTS = 24:64
```

일반적으로 응용프로그램의 쓰레드 수가 많을수록 더 많은 수의 아레나를 사용해야 성능이 향상된다. 하지만, 그렇다고 해서 너무 크게 설정하게 되면 힙(heap)영역의 단편화(fragmentation)가 발생, 비효율적인 메모리 사용으로 인해 프로세스의 사이즈가 지나치게 커져서 시스템 메모리 부족 현상이 발생할 수 있으므로 설정 시 면밀한 검토가 요구되는 사항이다.

예를 들어, 시스템의 메모리 리소스 자체가 적어 메모리 부족현상이 우려된다면 경우에 따라 단일프로세스 응용프로그램처럼 동작하도록 1:8 로 설정하기도 하며 성능저하 발생시 성능저하 원인이 메모리 요청(malloc/free)과 관련된 병목이라면 리소스가 허락하는 만큼 값을 늘리는 것이 일반적이다.

문자셋 관련 고려사항

ALTIBASE 5.3 버전 이전에는 문자셋과 관련하여 NLS_USE 라는 항목과 관련해 별도의 프로퍼티가 존재 하였으나 ALTIBASE 5.3 이후로 환경변수로 대체되었다.

이로 인해, ALTIBASE 5.3 버전 이상에서 DB의 문자셋을 US7ASCII가 아닌 KO16KSC5601 / MS949 / UTF8 등을 설정했을 경우에는 반드시 ALTIBASE_NLS_USE 라는 환경변수를 설정해야만 한다. 그렇지 않을 경우에는 기본설정인 US7ASCII로 접속하게 되어 입력 후 조회 시점에 알 수 없는 문자열 형태로 조회가 되게 된다.

터미널 환경

[환경변수] 섹션에서 언급한 문자셋 사용관련 고려사항과 더불어, 사용하는 터미널의 폰트(font) 역시 DB에 설정한 문자셋과 동일하게 설정하여야 한다.

멀티쓰레드 관련 패치

ALTIBASE는 단일 프로세스, 멀티쓰레드기반의 응용프로그램이다. 따라서, 멀티쓰레드 관련 HPUX 패치(patch)가 필요하다. 이 중에서도 "pthread library cumulative patch"는 성능과 직접적인 영향을 가지는 필수적인 요소로 반드시 확인되어야 한다.

현재 시스템의 패치리스트를 확인하는 방법은 아래와 같다.

```
# swlist -l patch |grep pthread
# PHCO_38955 1.0 pthread library cumulative patch
# PHKL_35709 1.0 pthread_cond_timedwait,hires timers,callout
# PHKL_37122 1.0 pthread_condvar_prio_boost patch
```

이 외에도 HPUX 에서 알려진 각종 문제를 피하기 위해 최신 패치 적용을 권고한다.

PHCO_33675 및 PHCO_34718 을 적용한 경우 추가 고려사항

HPUX11.23 에서 해당 패치를 적용한 경우, 응용 프로그램에서 공유 뮤텝스(mutex)를 더 오랫동안 보관하여 성능이 저하되는 현상이 HP를 통하여 보고되어 있다.

패치리스트를 확인하여 해당 패치가 존재한다면 다음과 같은 환경변수를 추가, 1 로 설정하여 시스템 성능을 복원하여야 한다.

환경변수	설명	비고
PTHREAD_SHARED_MUTEX_OLDSPIN	PHCO_33675, PHCO_34718 로 인한 성능저하 원복	HPUX 11.23 에서만 고려

요약

HPUX 운영체제에서 ALTIBASE의 안정적인 운영을 위해서는 커널파라미터 설정 및 사용자 환경 설정을 사전에 수행해야만 한다. 설정을 올바르게 수행하지 않은 경우, 시스템의 리소스가 충분함에도 불구하고 각각의 제한 값들에 의해 문제가 발생 할 수 있으므로 주의해야 한다.

설정의예

커널파라미터

아래 표를 참조, 적절한 커널파라미터를 설정하도록 한다. 참고로 HPUX에서는 리소스 제한 항목 중 일부를 커널파라미터를 변경으로 조절한다.

분류	커널파라미터	권장 값	비고
공유메모리	shmmni	500 이상	> shmseg
	shmseg	200 이상	-
세마포어	semmns	8192	-
	semmni	5029	-
	semmnu	5029	semmni
	semume	5029	semmni
파일캐시	dbc_min_pct	5 이하	-
	dbc_max_pct	20 이하	8G 이하 시스템에서는 20% 권장 8G 이상 시스템에서는 10% 권장
리소스 제한	maxdsiz	2G	data seg size에 대응 (32bit)
	maxdsiz_64bit	1T / 최대 약 4T	data seg size에 대응 (64bit)
	maxfiles	5029 이상	open files에 대응
	max_thread_proc	5029 이상	-
	nproc	6142	max user process 에 대응 >= semmnu + 4
	maxusers	124	HPUX 11.23 부터 사라짐

메모리DB를 공유메모리에 적재할 경우 추가 고려되어야 하는 요소는 [공유메모리] 섹션을 참조하도록 한다.

사용자 리소스 제한

아래 표를 참조, 가능하다면 모두 무제한으로 설정하도록 한다.

항목	설명	권장 값
data seg size(data)	프로세스 데이터영역의 최대크기	maxdsiz maxdsiz_64bit
file size (fsize)	생성되는 파일의 최대크기	unlimited
open files (nofiles)	하나의 프로세스가 동시에 접근 가능한 파일의 최대개수	maxfiles
max memory size (rss)	사용 가능한 메모리의 최대크기	unlimited
max user process	사용자당 생성 가능한 프로세스의 개수	nproc

사용자 환경변수

sh, bash, ksh의 경우, 환경설정파일을 이용한 필수 환경변수 설정예제는 아래와 같다. csh의 경우는 export 대신 setenv 와 같은 셸 명령어를 통해 선언한다.

```
#For Multi-Threaded Application
PTHREAD_FORCE_SCOPE_SYSTEM=ON; export PTHREAD_FORCE_SCOPE_SYSTEM;
PTHREAD_FAST_SHARED_OBJECTS=ON; export PTHREAD_FAST_SHARED_OBJECTS;
PTHREAD_DISABLE_HANDOFF=ON; export PTHREAD_DISABLE_HANDOFF;
PERF_ENABLE=1; export PERF_ENABLE;
_M_ARENA_OPTS=8:32; export _M_ARENA_OPTS;

#For ALTIBASE
ALTIBASE_HOME=/ALTIBASE; export ALTIBASE_HOME;
PATH=$ALTIBASE_HOME/bin:$PATH; export PATH;
ALTIBASE-NLS_USE=MS949; export ALTIBASE-NLS_USE;
SHLIB_PATH=$ALTIBASE_HOME/lib:$SHLIB_PATH;
export SHLIB_PATH; #For 32bit
LD_LIBRARY_PATH=$ALTIBASE_HOME/lib:$LD_LIBRARY_PATH;
export LD_LIBRARY_PATH; #For 64bit

#For Resource Limitation
ulimit -f unlimited #file size, fsize
ulimit -m unlimited #max memory size, rss
```

참고로 ksh의 경우, 환경변수가 미리 정의되지 않은 상태에서 환경변수를 사용하여 또 다른 환경변수를 정의 시 오류가 발생할 수 있다.

또한, “_M_ARENA_OPTS”는 단순히 기본값을 명시한 것으로 시스템에 따라 적절하게 고려가 되어야 한다. 상세한 내용은 [환경변수] 섹션의 “멀티쓰레드 응용프로그램 위한 설정(2)”을 참조하도록 한다.

이와 더불어, [멀티쓰레드 관련 패치] 섹션의 “PHCO_33675 및 PHCO_34718 을 적용한 경우 추가 고려사항”도 참조되어야 한다.



알티베이스㈜

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

솔루션센터

02-2082-1114
<http://support.altibase.com>

Copyright © 2000~2013 ALTIBASE Corporation. All Rights Reserved.

이 문서는 정보 제공을 목적으로 제공되며, 사전에 예고 없이 변경될 수 있습니다. 이 문서는 오류가 있을 수 있으며, 상업적 또는 특정 목적에 부합하는 명시적, 묵시적인 책임이 일체 없습니다. 이 문서에 포함된 ALTIBASE 제품의 특징이나 기능의 개발, 발표 등의 시기는 ALTIBASE 재량입니다. ALTIBASE는 이 문서에 대하여 관련된 특허권, 상표권, 저작권 또는 기타 지적 재산권을 보유할 수 있습니다.