## Informix Dynamic Server 관련 환경 변수

- 기본 환경 변수

| | |
|---|---|
| INFORMIXDIR | Informix 제품 설치 디렉토리 |
| PATH | $INFORMIXDIR/bin 을 기존 $PATH 에 추가 |
| INFORMIXSERVER | 데이터베이스 서버 이름 |

- 추가 환경 변수

| | |
|---|---|
| **GLS Locale** | |
| DB_LOCALE | 데이터베이스의 locale set 지정 변수<br>설정하지 않는 경우 en_us.8859-1 code set 을 적용 받음<br>KSC5601 완성형 한글 code set 을 사용하려면 ko_kr.ksc 로 설정<br>en_us.8859-1 로 설정하더라도 한글 처리는 가능하지만 column 명을 한글로 사용하려는 경우나 ordering 의 순서 등을 완성형에 맞도록 하려는 경우는 ko_kr.ksc 로 설정<br>en_us.8859-1 은 조합형의 글자로 모두 저장 가능 |
| SERVER_LOCALE | 데이터베이스 서버의 locale set 지정 변수<br>서버 초기화 단계에서 적용된 값이 계속 유지됨<br>설정 값은 DB_LOCALE 과 동일 |
| CLIENT_LOCALE | 데이터베이스 서버에 접근하는 클라이언트 프로그램의 locale code set 지정 변수<br>설정 값은 DB_LOCALE, SERVER_LOCALE 과 동일 |
| **Shared Library Path** | |
| LD_LIBRARY_PATH | SUN, LINUX 플랫폼에서 사용<br>$INFORMIXDIR/lib:$INFORMIXDIR/lib/esql 을 설정 |
| SHLIB_PATH | HP 플랫폼에서 사용 |
| LIBPATH | AIX 플랫폼에서 사용 |
| **Parallel Database Query(PDQ)** | |
| PDQPRIORITY | PDQ query 사용 여부 설정 및 PDQ resource 의 %<br>0 또는 설정하지 않는 경우 – PDQ 를 사용하지 않고 일반 query 로 적용<br>1 ~ 100 – PDQ 를 사용하는 경우 사용할 PDQ resource 의 %로 적용됨 |
| PSORT_NPROCS | 병렬 sort 처리에 사용되는 sort thread 의 개수 |
| **Tuning** | |
| OPTCOMPIND | Optimizer Hint 기능<br>0 – 색인을 이용한 scan<br>1 – Isolation level 이 repeatable read 인 경우엔 항상 색인을 이용한 scan 을 적용하고 그 외의 isolation level 인 경우에는 optimizer 가 판단 하도록 함<br>2 – Optimizer 가 판단하는 최적의 query path 를 따르도록 함<br>설정하지 않는 경우 default 값으로 2 가 적용됨 |
| FET_BUF_SIZE | 클라이언트 프로그램에서 사용하는 Fetch Buffer size<br>지정하지 않는 경우 4096 bytes 로 적용됨<br>최대 32767 bytes 까지 지정 가능 |
| **기타** | |
| ONCONFIG | Configuration 파일 이름을 default 값인 onconfig 가 아닌 다른 이름을 |

| | |
|---|---|
| | 사용하려는 경우 파일 이름 지정 |
| INFORMIXSQLHOSTS | sqlhosts 파일을 default 값인 $INFORMIXDIR/etc/sqlhosts 가 아닌 다른 이름의 파일을 사용하려는 경우 파일의 절대 경로 지정 |

**I/O Utilization Tuning Point**

- DBSPACE 구성

| |
|---|
| Root dbspace 와 Log dbspace, Data dbspace 는 반드시 분리 |
| 사용 가능한 디스크 구성에 따라서 Logical log 와 Physical log 를 각각의 dbspace 로 분리 |
| Grouping 이나 Sorting 등에 temporary 영역을 많이 사용하게 되는 경우, temporary dbspace 를 여러 개 생성 |

- LOG 구성

| |
|---|
| 전체 디스크 영역의 대략 20% 정도를 log 영역으로 할당 |
| Logical log 는 큰 크기로 적은 개수를 설정하는 것보다 작은 크기로 10 개 이상을 설정하는 것이 효과적 |
| Logical log 하나와 Physical log 의 크기 비율을 1:2 나 1:3 으로 구성 |
| Physical log 의 크기가 너무 작은 경우 check point 가 자주 발생하게 됨 |

- Configuration Parameter

| | |
|---|---|
| PHYSFILE | Physical log 의 크기 |
| | Physical log 가 75% 이상 차게 되면 check point interval 이 아니라도 check point 가 발생하게 되므로 모니터링을 통해서 physical log 의 크기를 조정 |
| | ***Log monitoring : onstat -l***<br>***Check point interval monitoring : onstat -m*** |
| NUMAIOVPS | Chunk 의 비동기방식 I/O 를 담당하는 VP(Virtual Processor)의 개수 |
| | Raw device 를 chunk 로 사용하는 경우에는 2~3 개로 설정<br>Cooked file 을 chunk 로 사용하는 경우에는 물리적인 디스크나 채널의 개수로 설정 |
| | ***AIO monitoring : onstat -g iof***<br>***onstat -g iog***<br>***onstat -g iov*** |
| CLEANERS | Check point 시에 write 요청을 담당하는 page cleaner thread 의 개수 |
| | 20 개 미만의 디스크를 사용하는 경우 1 개로 설정<br>20 개 이상 100 개 미만의 디스크를 사용하는 경우 디스크 2 개 당 1 개로 설정<br>100 개 이상의 디스크를 사용하는 경우 디스크 4 개당 1 개로 설정 |
| | Check point duration 이 길어 지는 경우 page cleaner thread 의 개수를 증가 |
| | ***Thread monitoring : onstat -g ath***<br>***Page cleaner thread activity monitoring : onstat -F*** |
| CKPTINTVL | Check point 발생 주기 |
| | Check point 가 진행되는 동안에는 데이터의 변경이 불가능하므로 check point duration 을 최소화 하는 것이 중요<br>Check point interval 값을 줄이게 되면 상대적으로 duration 시간이 줄어듦 |
| | ***Check point interval, duration monitoring : onstat –m***<br>***Check point 발생 횟수 monitoring : onstat -p*** |

| | |
|---|---|
| LRUS | 공유 메모리의 데이터 버퍼를 관리하는 Least Recently Used Queue 의 개수 |
| | 변경이 일어나지 않은 데이터 버퍼를 관리하는 FLRU queue 와 변경된 데이터 버퍼를 관리하는 MLRU queue 가 한 쌍으로 존재 |
| | LRU queue 의 개수가 증가되면 상대적으로 page cleaner thread 의 개수도 증가되는 것이 효과적 |
| | 하나의 LRU queue 쌍이 500~700 미만의 버퍼를 관리하도록 구성 |
| | ***LRU queue monitoring : onstat -R*** |
| LRU_MAX_DIRTY | Check point duration 을 최소화 하기 위해서 check point interval 사이에 LRU write 를 발생하도록 하는 시작 값으로서의 MLRU queue 사용 %<br>Ex) 80 으로 설정되어 있는 경우 MLRU queue 의 80%가 차게 되면 LRU write 가 발생하게 되므로 check point 시점에서는 write 할 데이터양이 줄어들게 됨 |
| | ***LRU queue monitoring : onstat -R***<br>***LRU write activity monitoring : onstat -F*** |
| LRU_MIN_DIRTY | LRU write 를 중단 하는 값으로서의 MLRU queue 사용 %<br>Ex) 70 으로 설정되어 있는 경우 LRU_MAX_DIRTY 값으로부터 LRU write 가 시작된 경우 MLRU queue 가 70%가 될 때 까지만 LRU write 수행 |
| | ***LRU queue monitoring : onstat -R***<br>***LRU write activity monitoring : onstat -F*** |
| LTXHWM | Long transaction 을 정의하는 logical log 사용 % |
| | ***Thread 별로의 logical log 사용 정도 및 long transaction 발생 횟수 monitoring : onstat -g tpf*** |
| LTXEHWM | Long transaction 발생 세션의 roll back 작업 동안 다른 세션의 작업을 보류하도록 하는 logical log 사용 % |
| RA_PAGES | Read Ahead(질의에 필요한 data page 또는 index page 를 shared memory 로 미리 읽어 들이는 기능)에 적용할 page 수<br>Default 값은 4pages, Read Ahead 기능을 사용하지 않는 경우 0 으로 설정 |
| | (BUFFERS * bp_fract) / (2 * large_queries) + 2<br>bp_fract : portion of data buffer to use read-ahead<br>large_queries : number of concurrent read-ahead queries |
| | ***Read Ahead monitoring : onstat -p*** |
| RA_THRESHOLD | Read Ahead 에 의해 읽혀진 page 를 memory 상에서 처리한 후 다시 Read Ahead 를 적용하는 시점<br>Read Ahead 에 의해서 처리되고 남은 page 수 |
| | (BUFFERS * bp_fract) / (2 * large_queries) – 2<br>bp_fract : portion of data buffer to use read-ahead<br>large_queries : number of concurrent read-ahead queries |

- Database Physical modeling 고려 사항

| |
|---|
| Database 는 별도의 Root dbspace 가 아닌 별도의 dbspace 에 생성 |
| 사용 빈도 높은 테이블에 대해서는 별도의 I/O 분산을 고려한 dbspace 분리 작업 필요 |
| 테이블 생성시 초기 extent size 와 next extent size 를 데이터 규모를 고려하여 지정<br>➢ Informix Dynamic Server 7.x 인 경우 index 를 별도의 dbspace 에 생성하지 않는 경우 extent size 계산에 index 영역도 추가<br>➢ Informix Dynamic Server 9.x 인 경우 index 는 항상 별도의 tablespace 에 생성되므로 테이블의 extent 는 실제 데이터만 고려해서 계산 |
| 가변적인 성격이 적은 문자열 컬럼에 대해서는 varchar 보다 char 를 사용하는 것이 효과적<br>varchar 를 사용하는 경우 maximum 값과 minimum 값을 동일하게 설정<br>(ex. Col1 varchar(30,30)) |

**CPU Utilization Tuning Point**

- Configuration Parameter

| | |
|---|---|
| NUMCPUVPS | CPU 작업을 전담하는 CPU vp(virtual processor : oninit process)의 개수 |
| | Client process 와 Informix Server process 가 동일한 시스템에서 실행되는 경우 : 물리적인 CPU 개수 – 1 로 설정<br>Informix Server process 만 실행되는 경우 : 물리적인 CPU 개수로 설정 |
| | ***VP Class monitoring : onstat -g glo***<br>***Threads monitoring : onstat -g ath*** |
| MULTIPROCESSOR | 시스템의 물리적인 CPU 의 개수가 2 개 이상이고 NUMCPUVP 를 2 개 이상으로 설정한 경우 1 로 설정, 그 외에는 0 로 설정 |
| SINGLE_CPU_VP | CPU vp 를 한 개만 사용하는 경우 1 로 설정<br>여러 개의 CPU vp 를 사용하는 경우 0 로 설정 |
| NUMAIOVPS | Disk I/O 요청을 전담하는 vp 의 개수 |
| | Kernel Asynchronous I/O 를 사용하는 경우 2~3 로 설정<br>Kernel Asynchronous I/O 를 사용하지 않는 경우 물리적인 Disk 개수나 I/O channel 개수로 설정 |
| | ***AIO VP activity monitoring : onstat -g iov***<br>***AIO VP queue monitoring : onstat -g ioq*** |
| NETTYPE | Client 로부터의 message 전달을 담당하는 poll thread 의 개수와 poll thread 가 실행되는 VP class 지정 parameter |
| | 1st position : poll thread 가 담당하는 protocol<br>connection type 에 따라서 각각의 poll thread 설정<br>sqlhosts 파일에 지정하는 protocol 과 동일하게 설정<br>2nd position : poll thread 의 개수<br>각 protocol 을 담당하는 poll thread 의 개수<br>listen thread 개수와 동일하게 설정하는 것이 효과적<br>message 처리 속도가 저하되는 경우 poll thread 의 개수 증가<br>3rd position : 하나의 poll thread 가 동시에 처리할 수 있는 message 의 개수<br>Shared memory connection 인 경우 동시 사용자 수로 제한됨<br>4th position : poll thread 가 실행되는 vp class (NET/CPU 중 선택)<br>CPU vp class 에서 실행되는 경우가 NET vp class 에서 실행되는 경우보다 속도가 빠르지만 하나의 CPU vp 에서 실행 될 수 있는 poll thread 의 개수는 한 개로 제한됨 |
| | ***Thread activity monitoring : onstat -g ath***<br>***Global network information monitoring : onstat -g ntu***<br>***Thread network information monitoring : onstat -g ntd*** |

- PDQ Configuration Parameter

| | |
|---|---|
| MAX_PDQPRIORITY | PDQ 질의를 요청하는 한 세션에게 최대 할당할 수 있는 Server 의 PDQ 자원의 % |
| | 한 세션이 할당 받는 자원 = PDQPRIORITY/100 * MAX_PDQPRIORITY/100 |
| | ***PDQ 질의 및 자원 monitoring : onstat -g mgm*** |
| DS_MAX_QUERIES | 동시에 실행 가능한 PDQ 질의 개수 |
| DS_TOTAL_MEMORY | PDQ 질의에 사용되는 서버쪽 메모리(Virtual portion 에서 할당)<br>DSS 환경인 경우 최대 시스템 메모리의 70%까지 설정 가능 |
| DS_MAX_SCANS | PDQ 질의에 사용되는 scan thread 개수 |

  - PDQ 질의를 사용하기 위해서는 세션의 환경변수로 PDQPRIORITY 를 설정하거나 SQL 문 “SET PDQPRIORITY ~”을 통해서 사용 여부를 명시해야 함(Default 값은 PDQ 기능을 사용하

지 않는 일반 질의로 처리됨)

- 시스템 자원을 많이 필요로 하는 배치성격의 질의가 아니더라도 Index 생성 작업이나 많은 양의 Data loading 작업 등에 PDQ 기능을 사용하면 질의 수행 속도를 높일 수 있음

**MEMORY Utilization Tuning Point**

- Resident Portion Parameter

| | |
|---|---|
| BUFFERS | Shared memory 상의 data cashing pool 크기를 지정하는 값<br>Buffer 의 개수로 지정, Buffer 하나의 크기는 page 크기와 동일(page 크기는 onstat -b 로 확인) |
| | 시스템 메모리의 30% 정도를 초기값으로 설정, 이 후에 read cash %를 보고 98% 이상이 될 때까지 값을 증가 |
| | ***Memory cash% monitoring : onstat -p*** |
| LOCKS | 데이터베이스의 동시성 제어를 위해 사용되는 Shared level lock, Exclusive level lock, Update level lock 등의 개수 |
| | 최소 2000 개, 최대 8 백만개<br>onstat -p 를 통해 lock overflow 가 일어나지 않는 지 확인 후 최대한 여유있게 설정(lock 한 개가 차지하는 메모리 크기가 작으므로 전체 메모리 크기에 큰 영향을 주지는 않음) |
| | ***Lock overflow monitoring : onstat -p***<br>***Lock 사용 상태 monitoring : onstat -k*** |
| PHYSBUFF | 디스크의 Physical log 로 flush 되는 before image 를 모아두는 메모리 상의 buffer 크기 |
| | Default 값은 32KB<br>Buffer 의 크기가 작으면 디스크의 Physical log 로 자주 flush 가 일어나므로 I/O 성능이 저하됨 |
| | ***Physical log / log buffer monitoring : onstat -l*** |
| LOGBUFF | 디스크의 Logical log 로 flush 되는 log history 를 모아두는 메모리 상의 buffer 크기 |
| | Default 값은 32KB<br>Buffer 의 크기가 작으면 디스크의 Logical log 로 자주 flush 가 일어나므로 I/O 성능이 저하됨 |
| | ***Logical log / log buffer monitoring : onstat -l*** |

- Virtual Portion Parameter

| | |
|---|---|
| SHMVIRTSIZE | 초기에 할당되는 Virtual Portion 의 크기<br>각 세션이 필요로 하는 virtual portion 의 크기를 monitoring 을 통해서 확인 후에 적절한 크기로 설정 |
| | ***Shared memory segment monitoring : onstat -g seg*** |
| SHMADD | 초기에 할당된 virtual portion 이 부족한 경우 추가로 할당되는 segment 의 크기<br>자동으로 필요한 만큼의 segment 가 계속 할당되지만 segment 의 개수가 너무 많아지게 되면 서버의 성능이 저하되므로 필요한 만큼의 크기를 SHMVIRTSIZE 로 지정하는 것이 효과적임 |