MYSQL 최적화 방법 설명서

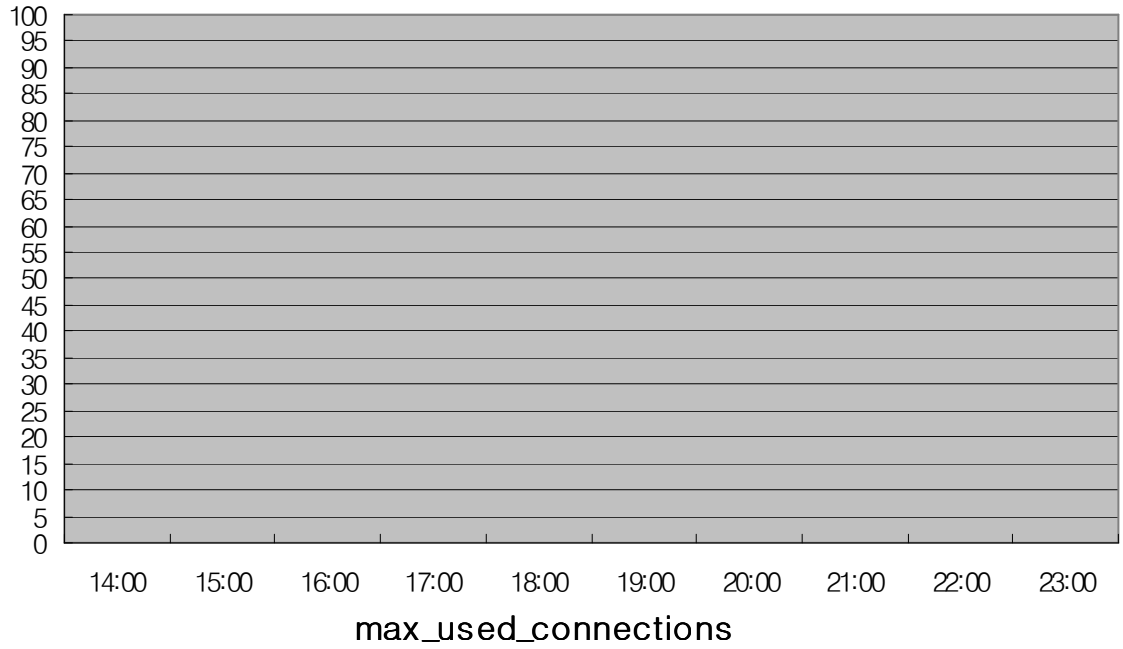
(1) 동시 접속자수 설정

1) 관련 값

- max_used_connections : 동시 접속자 최대 건수 (읽기전용)
- connections : 연결 시도된 총 수치
- max_connections : 동시 접속자 수(최적화 시켜야 할 값)
- max_connect_error : 지정된 이상의 연결장애가 발생하면 접속 호스트를 블락시킴.
- max_user_connections = 0 (기본값(no limit) : 하나의 유저당 할당된 컨넥션 수)

2) 최적화 방법

- max_used_connections 값을 모니터링 해서 동시에 사용한 최대 사용자수를 확인한다.



3) 최적화 결과

- max_connections =
- max_connect_error =

4) 참고

max_connect_error - DB 연결 에러가 지정한 값 이상으로 발생할 경우 접속하는 해당 호스트(웹서버와 같은 PC에 있는 경우에는 웹서버를 가리킴)을 블락시켜버린다. 이 경우에는 MYSQL 서버에서 mysqladmin flush-hosts 를 해줘야 하는데 실제 실행키져보면 블락이 바로 해제되지 않는다. 이 값은 보통 max_connections 값과 동일하게 설정하거나 크게 설정한다.

(2) 최대 open 테이블 수 설정

1) 설명: 모든 쓰레드에서 오픈할 수 있는 DB테이블 총 개수

2) 관련 값

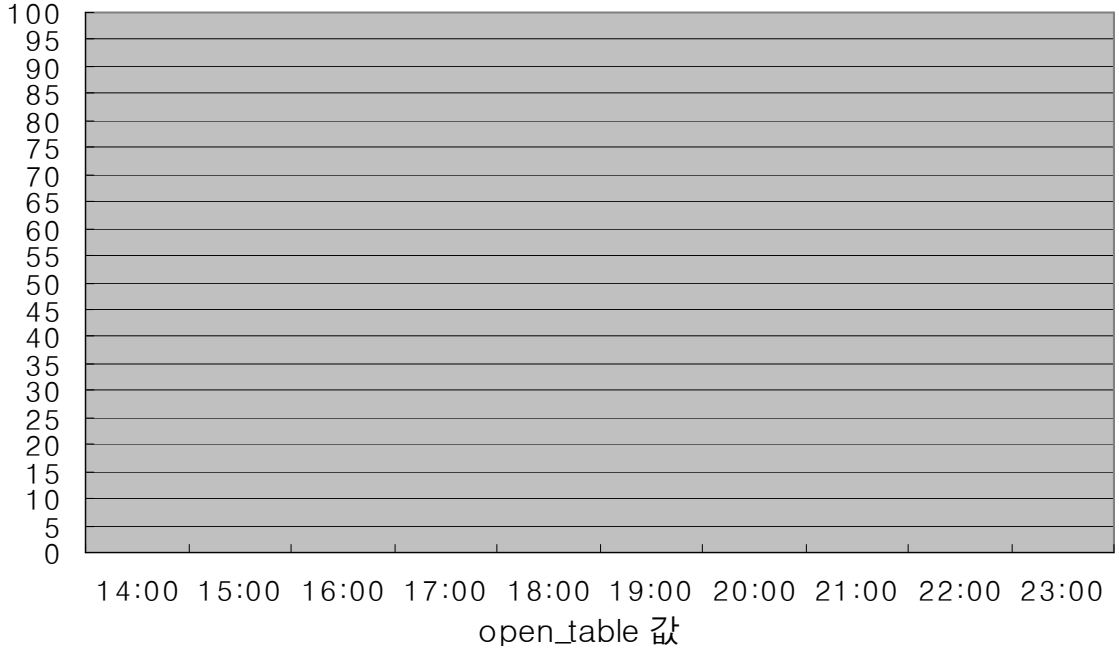
- max_connections
- open_tables : 현재 오픈된 테이블 수(읽기전용)
- open_files : 현재 오픈된 파일 수 (읽기전용)
- opened_tables : 지금까지 오픈된 페이블 수 (읽기전용)
- table_cache = 350 (기본값)
- open_files_limit = 0 (기본값: max_connections * 5 혹은 max_connections + table_cache*2)

3) 최적화 방법

- table_cache -

방법1: opened_tables 을 모니터링해서 이 값이 크면 table_cache값을 증가시킬 필요가 있다.

방법2: open_table 값을 체크함으로서 테이블캐쉬값을 증가시키는것이 필요한 지 체크할 수 있다.
보통 이 값을 max_connections 의 1.5배 정도 크기로 설정하는 것이 좋다고 한다.



3) 최적화 결과

- table_cache =

(3) timeout 시간 설정

1) 설명: 각종 실행 초과 값을 지정한다.

2) 관련 값

- max_execution_time = 2400 (php.ini파일 , php스크립트 최대 실행시간)
- mysql.connect_timeout = 10 (php.ini파일 , db연결 타임아웃 시간)
- connect_timeout = 5 (초과하면 db접속에러가 남, 에러발생은 aborted_connections 에 기록됨)
- wait_timeout : 1800 (쿼리실행 타임아웃 값. mysql 접속해제시 접속 링크가 컨넥션 풀에 계속 남아있도록 하는 것을 방지하기 위해서 설정하는 값이다. 하지만 실제로는 접속해제가 되지 않고 계속 메모리에 남아있는다. 참고로 이 값은 php.ini에서 mysql.allow_persistent 값이 true로 설정되어 있는 경우에만 유효하면 에러발생은 aborted_clients 에 기록된다. 실제로 해제되지 않기 때문에 의미가 없다.)

3) 최적화 방법

- 웹서버와 DB서버가 같이 PC에 있는 경우에는 영구적인 접속이 비영구적인 접속에 비해 어떠한 기능적인 향상을 주지 못할 수 있으며, 접속자 수가 많지 않은 되도 불구하고 " **Too many connections** " 오류를 발생할 수 있기 때문에 php에서 영구적인 접속을 꺼두는 것이 좋다.

4) 최적화 결과

- connect_timeout = 5
- mysql.allow_persistent = off (php.ini 파일)

(4) 인덱스 모니터링

1) 설명: 인덱스를 제대로 사용하고 있는지 확인할 수 있다.

2) 관련 값

- handler_read_first : 인덱스로부터 읽혀진 처음 엔트리 수, 이 값이 높으면 서버는 많은 full index scans를 하고 있다는 것을 의미한다. 예를 들어 select col1 from foo 는 col1은 인덱스되었다는 것을 추정한다.
- handler_read_rnd : 이 값이 크면 모든 테이블을 스캔하는 많은 쿼리가 있다거나 key를 적절히 사용하지 않는 조이들이 있을지 모른다.

(5) 인덱스 버퍼 설정

1) 설명: 인덱스블럭에 사용되어지는 버퍼사이즈

2) 관련 값

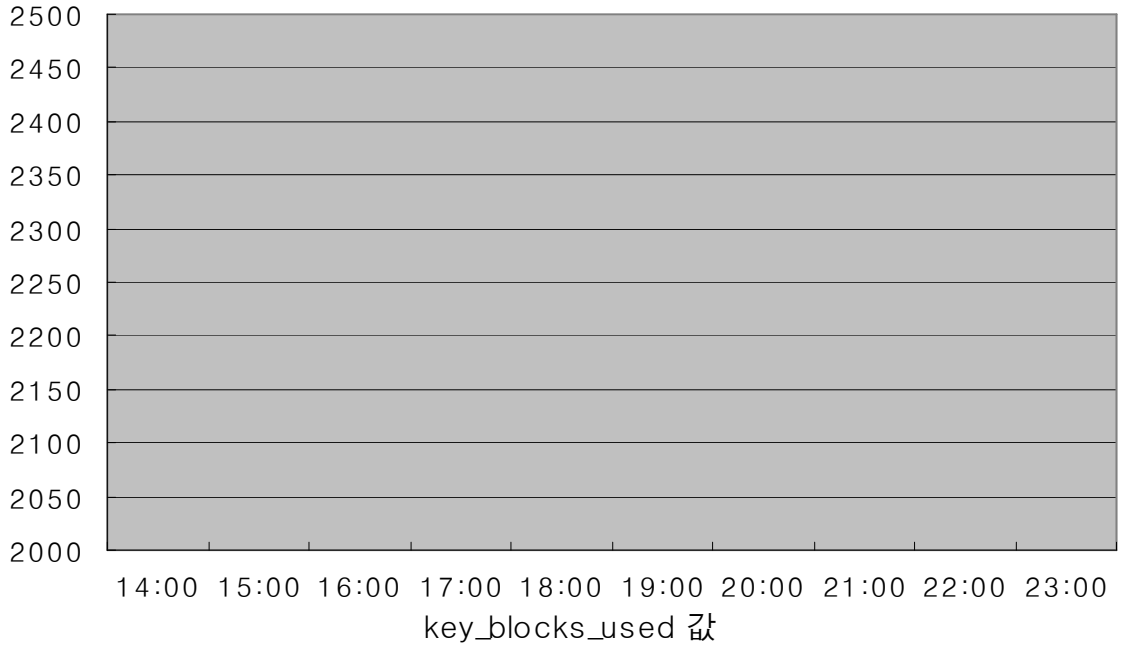
- key_buffer_size = 8,388,600 M
- key_blocks_used = 2947 (읽기전용)
- key_read_requests = 173351 (읽기전용)
- key_reads = 2226 (읽기전용)
- key_wirte_requests =1027 (읽기전용)
- key_writes =772 (읽기전용)

3) 최적화 방법

- key_reads/key_read_requests 율이 보통 0.01보다 작아야 한다.
  update ,delete 를 대부분 사용한다면 key_write/key_write_requests 가 1에 가까워지는게 일반적이고 동시에 update을 많이 하거나 delay_key_write 를 사용한다면 이 비율은 작아진다.
  key_blocks_used * 1024 면 충분하다.

3) 최적화 결과

- key_buffer_size > 3.017728 M (key_blocks_used * 1024) 설정하면 된다.
  현재 설정된 값을 그대로 사용한다.



(6) SQL 정렬 버퍼 설정

1) 설명: 더 빠른 order by group by 수행을 위해서 이 값들을 증가시켜라

2) 관련 값

- max_connections = 200
- sort_buffer = 2.097144 M
- record_buffer = 131.072 K (연속적인 데이타 스캔)
- record_rnd_buffer = 131.072 K (정렬된 순서로 데이터 스캔)
- handler_read_key : 키가 존재하는 ros를 일는 요청수, 이 값이 높으면 테이블이 적절히 인덱스 되었다는 되었다는 것을 말해준다.
- max_sort_length = 1024 (BLOB,TEXT 테이타를 정렬할 때 사용됨)
- sort_rows : 정렬된 레코드 수
- sort_scan : 테이블 스캔에 의해 행해진 정렬 수

3) 최적화 방법

- good_sort_buffer_size = (sort_buffer + record_buffer) * max_connections 라고 정의한다. 이 값이 실메모리크기보다 크지 않도록 한다.

4) 최적화 결과

- 현재 good_sort_buffer_size 값이 445.343200 M 이므로 실메모리값(2G)보다 크지 않으므로 현재 설정된 값을 그대로 사용하면 된다.

(7) SQL 쿼리 모니터링

1) 설명: 서버에 보내는 SQL쿼리에 문제는 없는지 확인할 수 있다.

2) 관련 값

- Questions : 서버에 보낸 쿼리수
- query_buffer_size = 0 (no limit : 쿼리버퍼의 초기할당, 대부분의 쿼리가 길다면 이 값을 증가시킨다.)
- long_query_time = 10 (일정 시간 이상 실행되는 쿼리 기준 시간)
- log_long_queries : long_query_time 이상 실행된 쿼리를 로그에 남김
- slow_queries : long_query_time 보다 더 많은 시간이 걸리는 쿼리수. 이 값은 0이면 좋다.
- table_locks_immediate : 즉시 획득된 테이블 lock 시간
- table_locks_waited : 즉시 획득되지 않고 기다림이 필요한 테이블 lock 시간
  이 것이 높아지면 성능에 문제가 있으므로 먼저 쿼리를 최적화 시키고 , 테이블을 분산시키거나 복제를 사용해야 한다.

(8) 테이블 JOIN

1) 관련 값

- select_full_join : 키없이 조인된 수(0이 되어야 한다.)
- join_buffer_size = 131072 (인덱스를 사용하지 않는 full조인에 사용되는 버퍼사이즈)
- max_join_size = 4.294967295G (where절이 없고 오래걸리는 조인과 많은 레코드를 반환하는 유저가 있으면 적절히 설정)

(9) 테이블 생성

1) 관련 값

- tmp_table_size = 33.554432 M (메모리안에 임시테이블이 이 사이즈를 초과하면 자동적으로 임시테이블을 디스크에 저장한다.)
- created_tmp_tables : 메모리에 생성된 임시테이블 생성수
- created_tmp_disk_tables :디스크에 존재하는 임시테이블 수
- created_tmp_files : 임시파일 생성수

2) 최적화 방법

- created_tmp_disk_tables 값이 크면 tmp_table_size 값을 증가시켜야 한다.

3) 최적화 결과

- tmp_table_size = 100M

(10) 쓰레드 관련 모니터링

1) 관련 값

- thread_cache_size = 0 (컨넥션이 이루어질 때 캐쉬에 있는 쓰레드를 사용하지 말고 새루운 쓰레드를 생성하여 사용하도록 강제)
- thread_stack = 65536
- threads_cached : 쓰레드 캐쉬에 있는 쓰레드 수
- threads_connected : 현재 오픈된 연결수
- threads_created : 연결을 다루기 위해 생성된 쓰레드 수
- threads_running : sleeping하지 않는 쓰레드 수
- slow_launch_time : 쓰레드 생성시간이 이 값보다 길면 slow_launch_threads 수치는 증가
- slow_launch_threads
- delayed_insert_threads : 사용중인 insert_handler thread 가 지연되고 있는 수
- delayed_writes : insert delayed 로 쓰여진 rows 수
- delayed_errors : 에러로 쓰여진 rows수
- handler_delete : 테이블로부서 지워진 rows수

2) 최적화 방법

- thread_created 값이 크면 thread_cache_size 값을 증가시켜야 한다.
- 매우 바쁜 MYSQL 서버가 아니면 0 또는 2값 정도면 충분하다.
- thread_created / connections 율이 1% 정도면 이 값을 증가시킬 필요가 있다.

3) 최적화 결과

- thread_cache_size = 2

(11) 보안관련 설정

1) 관련된 값

- safe_show_database = OFF(유저에게 권한이 없는 데이터베이스에 대해 보여줄지 말지에 대한 설정)
- skip_show_database = OFF(process_priv권한이 없는 유저에게 show databases 를 못하게 한다.)

(12) MYSQL데몬 구동 방법

net stop mysql

mysqld-nt --default-character-set=euc_kr -O back_log=50 -O max_connections=200 -O max_connect_err=300 -O table_cache=350 -O tmp_table_size=100M -O wait_timeout=30 -O interactive_timeout=1800 -O thread_cache_size=2 &

기타.

MYSQLADMIN 툴을 이용해서 서비스 모니터링하는 방법 소개

- MYSQL 서버가 살아있는지 확인하는 방법

  mysqladmin -u root -p ping

- MYSQL 쓰레드 목록보기

  mysqladmin -u root -p proc stat(=processlist)

- MYSQL 서버환경설정 확인

  mysqladmin -u root -p variables(※mysql>show valiables)

  mysqladmin -u root -p extended-status(※mysql>show stauts)

- (팁) 배치파일로 연속해서 확인하는 방법은 다음과 같다.

  checking.bat이라는 파일을 만들어서 다음과 같이 작성한다.

```
:loop
mysqladmin -u root -p proc stat(=processlist)
pause
go loop
```

작 성 자 : Lim min taek(c_min2000@yahoo.co.kr)

작성날짜 : 2005-10-08