Mysql Connection my.cnf config file and explanations

[crayon-66ed95265fce3719704907/]

……>

query_cache_size:

MySQL 4 provides one feature that can prove very handy – a query cache. In a situation where the database has to repeatedly run the same queries on the same data set, returning the same results each time, MySQL can cache the result set, avoiding the overhead of running through the data over and over and is extremely helpful on busy servers.

……>

key_buffer_size:

The value of key_buffer_size is the size of the buffer used with indexes. The larger the buffer, the faster the SQL command will finish and a result will be returned. The rule-of-thumb is to set the key_buffer_size to at least a quarter, but no more than half, of the total amount of memory on the server. Ideally, it will be large enough to contain all the indexes (the total size of all .MYI files on the server).

A simple way to check the actual performance of the buffer is to examine four additional variables: key_read_requests, key_reads, key_write_requests, and key_writes.

If you divide the value of key_read by the value of key_reads_requests, the result should be less than 0.01. Also, if you divide the value of key_write by the value of key_writes_requests, the result should be less than 1.

……>

table_cache:

The default is 64. Each time MySQL accesses a table, it places it in the cache. If the system accesses many tables, it is faster to have these in the cache. MySQL, being multi-threaded, may be running many queries on the table at one time, and each of these will open a table. Examine the value

of open_tables at peak times. If you find it stays at the same value as your table_cache value, and then the number of opened_tables starts rapidly increasing, you should increase the table_cache if you have enough memory.

……>

sort_buffer:

The sort_buffer is very useful for speeding up myisamchk operations (which is why it is set much higher for that purpose in the default configuration files), but it can also be useful everyday when performing large numbers of sorts.

……>

read_rnd_buffer_size:

The read_rnd_buffer_size is used after a sort, when reading rows in sorted order. If you use many queries with ORDER BY, upping this can improve performance. Remember that, unlike key_buffer_size and table_cache, this buffer is allocated for each thread. This variable was renamed from record_rnd_buffer in MySQL 4.0.3. It defaults to the same size as the read_buffer_size. A rule-of-thumb is to allocate 1KB for each 1MB of memory on the server, for example 1MB on a machine with 1GB memory.

……>

thread_cache:

If you have a busy server that’s getting a lot of quick connections, set your thread cache high enough that the Threads_created value in SHOW STATUS stops increasing. This should take some of the load off of the CPU.

……>

tmp_table_size:

“Created_tmp_disk_tables” are the number of implicit temporary tables on disk created while executing statements and “created_tmp_tables” are memory-based. Obviously it is bad if you have to go to disk instead of memory all the time.

Additional reference material: