链滴

mysql 5.7 配置项最详细的解释

作者：centrexzj
原文链接：https://ld246.com/article/1536649376592
来源网站：链滴

mysql 5.7配置项最详细的解释

```
[client]
#password=88888888
socket=/data/var/mysql/mysql.sock

[mysqld_safe]
pid-file=/data/var/mysql/mysqld.pid
log-error = /var/lib/mysql/log/mysql-error.log

[mysql]
socket=/data/var/mysql/mysql.sock

[mysqld]
user = mysql
port = 31306
datadir = /data/var/mysql
socket=/data/var/mysql/mysql.sock
symbolic-links=0
########basic settings########
server-id = 11
#bind_address = 10.166.224.32
autocommit = 1
character_set_server=utf8mb4
skip_name_resolve = 1
max_connections = 800
max_connect_errors = 100
transaction_isolation = READ-COMMITTED
explicit_defaults_for_timestamp = 1
join_buffer_size = 128M
tmp_table_size = 128M
tmpdir = /dev/shm
max_allowed_packet = 16M
sql_mode = "STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION,NO_ZERO_DATE,NO_ZERO_
N_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER"
interactive_timeout = 60
wait_timeout = 60
read_buffer_size = 16M
read_rnd_buffer_size = 32M
sort_buffer_size = 32M
########log settings########
log_error = /var/lib/mysql/log/mysql-error.log
slow_query_log = 1
slow_query_log_file = /var/lib/mysql/log/mysql-slow.log
log_queries_not_using_indexes = 1
log_slow_admin_statements = 1
log_slow_slave_statements = 1
log_throttle_queries_not_using_indexes = 10
expire_logs_days = 90
long_query_time = 1
min_examined_row_limit = 100
########replication settings########
#master_info_repository = TABLE
```

```
#relay_log_info_repository = TABLE
log_bin = /var/lib/mysql/log/mysql-bin
#sync_binlog = 4
gtid_mode = on
enforce_gtid_consistency = 1
#log_slave_updates
binlog_format = row
#relay_log = /var/lib/mysql/log/mysql-relay.log
#relay_log_recovery = 1
#binlog_gtid_simple_recovery = 1
#slave_skip_errors = ddl_exist_errors
########innodb settings########
innodb_page_size = 16K
innodb_buffer_pool_size = 4G
#innodb_buffer_pool_instances = 8
#innodb_buffer_pool_load_at_startup = 1
#innodb_buffer_pool_dump_at_shutdown = 1
#innodb_lru_scan_depth = 2000
innodb_lock_wait_timeout = 5
#innodb_io_capacity = 4000
#innodb_io_capacity_max = 8000
innodb_flush_method = O_DIRECT
innodb_log_group_home_dir = /var/lib/mysql/log/redolog/
innodb_undo_directory = /var/lib/mysql/log/undolog/
innodb_undo_logs = 128
innodb_undo_tablespaces = 0
innodb_flush_neighbors = 1
#innodb_log_file_size = 4G
#innodb_log_buffer_size = 16M
#innodb_purge_threads = 4
innodb_large_prefix = 1
innodb_thread_concurrency = 64
#innodb_print_all_deadlocks = 1
#innodb_strict_mode = 1
innodb_sort_buffer_size = 64M
########semi sync replication settings########
#plugin_dir=/var/lib/mysql/lib/plugin
#plugin_load = "rpl_semi_sync_master=semisync_master.so;rpl_semi_sync_slave=semisync_sla
e.so"
#loose_rpl_semi_sync_master_enabled = 1
#loose_rpl_semi_sync_slave_enabled = 1
#loose_rpl_semi_sync_master_timeout = 5000

[mysqld-5.7]
#innodb_buffer_pool_dump_pct = 40
innodb_page_cleaners = 4
#innodb_undo_log_truncate = 1
#innodb_max_undo_log_size = 2G
#innodb_purge_rseg_truncate_frequency = 128
#binlog_gtid_simple_recovery=1
log_timestamps=system
#transaction_write_set_extraction=MURMUR32
#show_compatibility_56=on
```

详细解释

[client]

1. mysql默认密码

#password=88888888

2. mysql以socket方式运行的sock文件位置

socket=/data/var/mysql/mysql.sock

[mysqld_safe]

3. 错误日志位置

log-error=/var/log/mysqld.log

4. 进程id

pid-file=/var/run/mysqld/mysqld.pid

[mysql]

5. mysql以socket方式运行的sock文件位置

socket=/data/var/mysql/mysql.sock

[mysqld]

5. mysql以什么用户运行

user = mysql

6. mysql运行在哪个端口

port = 31306

7. mysql的数据目录

datadir = /data/var/mysql/

8. mysql以socket方式运行的sock文件位置

socket=/data/var/mysql/mysql.sock

9. 是否支持符号链接，即数据库或表可以存储在my.cnf中指定datadir之外的分区或目录，为0不开启

symbolic-links=0

######basic settings########

10. mysql的服务器分配id，在启用主从和集群的时候必须指定，每个节点必须不同

server-id = 11

11. mysql监听的ip地址，如果是127.0.0.1，表示仅本机访问

#bind_address = 10.166.224.32

12. 数据修改是否自动提交，为0不自动提交

autocommit = 1

13. 服务器使用的字符集

character_set_server=utf8mb4

14. 禁用DNS主机名查找，启用以后用内网地址向mysqlslap请求响应快了一半

skip_name_resolve = 1

15. mysql最大连接数

max_connections = 800

16.某台host连接错误次数等于max_connect_errors（默认10），主机'host_name'再次尝试时被屏。可有效反的防止dos攻击

max_connect_errors = 1000

16. 数据库事务隔离级别

transaction_isolation = READ-COMMITTED

- READ-UNCOMMITTED(读取未提交内容)级别
- READ-COMMITTED（读取提交内容）
- REPEATABLE-READ(可重读)
- SERIERLIZED(可串行化)
- 默认级别REPEATABLE-READ

17.mysql中TIMESTAMP类型和其他的类型有点不一样(在没有设置explicit_defaults_for_timestamp 1的情况下)

explicit_defaults_for_timestamp = 1

18. 当我们的join是ALL,index,rang或者Index_merge的时候使用的buffer。 实际上这种join被称为FLL JOIN

join_buffer_size = 128M

19. 规定了内部内存临时表的最大值，每个线程都要分配。（实际起限制作用的是tmp_table_size和mx_heap_table_size的最小值。）如果内存临时表超出了限制，MySQL就会自动地把它转化为基于磁的MyISAM表，存储在指定的tmpdir目录下

tmp_table_size = 128M

20. 保存临时文件的目录

tmpdir = /dev/shm/mysql-tmp/

21. mysql最大接受的数据包大小

max_allowed_packet = 16M

22. sql_mode 模式，定义了你MySQL应该支持的sql语法，对数据的校验等等，限制一些所谓的‘合法’的操作

sql_mode = "STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION,NO_ZERO_DATE,NO_ZERO_N_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER"

23. 服务器关闭交互式连接前等待活动的秒数。交互式客户端定义为在mysql_real_connect()中使用CLENT_INTERACTIVE选项的客户端

interactive_timeout = 60

24. 服务器关闭非交互连接之前等待活动的秒数，在线程启动时，根据全局wait_timeout值或全局interactive_timeout值初始化会话wait_timeout值，取决于客户端类型(由mysql_real_connect()的连接项CLIENT_INTERACTIVE定义)

wait_timeout = 60

25. 读入缓冲区的大小，将对表进行顺序扫描的请求将分配一个读入缓冲区，MySQL会为它分配一段存缓冲区

read_buffer_size = 16M

26. 随机读缓冲区大小，当按任意顺序读取行时（列如按照排序顺序）将分配一个随机读取缓冲区，行排序查询时，MySQL会首先扫描一遍该缓冲，以避免磁盘搜索，提高查询速度

read_rnd_buffer_size = 32M

27. 是一个connection级参数，在每个connection第一次需要使用这个buffer的时候，一次性分配设的内存

sort_buffer_size = 32M

######log settings########

28. 错误日志位置

#log_error = /data/local/mysql-5.7.19/log/mysql-error.log

29. 是否开启慢查询日志收集

slow_query_log = 1

30. 慢查询日志位置

slow_query_log_file = /data/local/mysql-5.7.19/log/mysql-slow.log

31. 是否记录未使用索引的语句

log_queries_not_using_indexes = 1

32. 慢查询也记录那些慢的optimize table，analyze table和alter table语句

log_slow_admin_statements = 1

33. 记录由Slave所产生的慢查询

log_slow_slave_statements = 1

34. 设定每分钟记录到日志的未使用索引的语句数目，超过这个数目后只记录语句数量和花费的总时间

log_throttle_queries_not_using_indexes = 10

35. 日志自动过期清理天数

expire_logs_days = 90

36. 设置记录慢查询超时时间

long_query_time = 1

37. 查询检查返回少于该参数指定行的SQL不被记录到慢查询日志

min_examined_row_limit = 100

######replication settings########

38. 从机保存主节点信息方式，设成file时 会生成master.info 和 relay-log.info2个文件，设成table信息就会存在mysql.master_slave_info表中。不管是设置的哪种值，都不要移动或者编辑相关的文和表

#master_info_repository = TABLE

39. 用于保存slave读取relay log的位置信息，可选值为“FILE”、“TABLE”，以便crash重启后继恢复

#relay_log_info_repository = TABLE

40. binlog的保存位置，不能指定确定的文件名如mysql-bin.log，只能指定位置和前缀，会生成以前为开头的一系列文件

log_bin = /data/local/mysql-5.7.19/log/mysql-bin

40. 这个参数是对于MySQL系统来说是至关重要的，他不仅影响到Binlog对MySQL所带来的性能损，而且还影响到MySQL中数据的完整性。对于“sync_binlog”参数的各种设置的说明如下：

● sync_binlog=0，当事务提交之后，MySQL不做fsync之类的磁盘同步指令刷新binlog_cache中的息到磁盘，而让Filesystem自行决定什么时候来做同步，或者cache满了之后才同步到磁盘。

● sync_binlog=n，当每进行n次事务提交之后，MySQL将进行一次fsync之类的磁盘同步指令来将bin og_cache中的数据强制写入磁盘。

● 在MySQL中系统默认的设置是sync_binlog=0，也就是不做任何强制性的磁盘刷新指令，这时候的能是最好的，但是风险也是最大的。因为一旦系统Crash，在binlog_cache中的所有binlog信息都会丢失。而当设置为“1”的时候，是最安全但是性能损耗最大的设置。因为当设置为1的时候，即使系Crash，也最多丢失binlog_cache中未完成的一个事务，对实际数据没有任何实质性影响。从以往经和相关测试来看，对于高并发事务的系统来说，“sync_binlog”设置为0和设置为1的系统写入性能距可能高达5倍甚至更多。

#sync_binlog = 4

41. 启用gtid类型，否则就是普通的复制架构

gtid_mode = on

42. 强制GTID的一致性

enforce_gtid_consistency = 1

43. slave更新是否记入日志，在做双主架构时异常重要，影响到双主架构是否能互相同步

#log_slave_updates

44. binlog日志格式，可选值“MIXED”、“ROW”、“STATEMENT”，在5.6版本之前默认为“STATEMENT”，5.6之后默认为“MIXED”；因为“STATEMENT”方式在处理一些“不确定”性的方时会造成数据不一致问题，我们建议使用“MIXED”或者“ROW”

binlog_format = row

45. 从机保存同步中继日志的位置

#relay_log = /data/local/mysql-5.7.19/log/mysql-relay.log

46. 当slave从库宕机后，假如relay-log损坏了，导致一部分中继日志没有处理，则自动放弃所有未执的relay-log，并且重新从master上获取日志，这样就保证了relay-log的完整性

#relay_log_recovery = 1

47. 这个参数控制了当mysql启动或重启时，mysql在搜寻GTIDs时是如何迭代使用binlog文件的。个选项设置为真，会提升mysql执行恢复的性能。因为这样mysql-server启动和binlog日志清理更快

#binlog_gtid_simple_recovery = 1

48. 跳过指定error no类型的错误，设成all 跳过所有错误

#slave_skip_errors = ddl_exist_errors

######innodb settings########

49. innodb每个数据页大小，这个参数在一开始初始化时就要加入my.cnf里，如果已经创建了表，再改，启动MySQL会报错

innodb_page_size = 16K

50. 缓存innodb表的索引，数据，插入数据时的缓冲，专用mysql服务器设置的大小： 操作系统内存70%-80%最佳

innodb_buffer_pool_size = 4G

51. 可以开启多个内存缓冲池，把需要缓冲的数据hash到不同的缓冲池中，这样可以并行的内存读写

#innodb_buffer_pool_instances = 8

52. 默认为关闭OFF。如果开启该参数，启动MySQL服务时，MySQL将本地热数据加载到InnoDB缓池中

#innodb_buffer_pool_load_at_startup = 1

53. 默认为关闭OFF。如果开启该参数，停止MySQL服务时，InnoDB将InnoDB缓冲池中的热数据保到本地硬盘

#innodb_buffer_pool_dump_at_shutdown = 1

54. 根据 官方文档 描述，它会影响page cleaner线程每次刷脏页的数量， 这是一个每1秒 loop一次线程

#innodb_lru_scan_depth = 2000

55. 事务等待获取资源等待的最长时间，超过这个时间还未分配到资源则会返回应用失败；参数的时单位是秒

innodb_lock_wait_timeout = 5

56. 这两个设置会影响InnoDB每秒在后台执行多少操作. 大多数写IO(除了写InnoDB日志)是后台操作的 如果你深度了解硬件性能(如每秒可以执行多少次IO操作),则使用这些功能是很可取的,而不是让它闲着

#innodb_io_capacity = 4000

#innodb_io_capacity_max = 8000

57. 默认值为 fdatasync. 如果使用 硬件RAID磁盘控制器, 可能需要设置为 O_DIRECT. 这在读取InnoB缓冲池时可防止“双缓冲(double buffering)”效应,否则会在文件系统缓存与InnoDB缓存间形成2副本(copy). 如果不使用硬件RAID控制器,或者使用SAN存储时, O_DIRECT 可能会导致性能下降

#innodb_flush_method = O_DIRECT

58. innodb重做日志保存目录

#innodb_log_group_home_dir = /data/local/mysql-5.7.19/log/redolog/

59. innodb回滚日志保存目录

#innodb_undo_directory = /data/local/mysql-5.7.19/log/undolog/

60. undo回滚段的数量， 至少大于等于35，默认128

#innodb_undo_logs = 128

61. 用于设定创建的undo表空间的个数，在mysql_install_db时初始化后，就再也不能被改动了；默值为0，表示不独立设置undo的tablespace，默认记录到ibdata中；否则，则在undo目录下创建这多个undo文件，例如假定设置该值为4，那么就会创建命名为undo001~undo004的undo tablespac文件，每个文件的默认大小为10M。修改该值会导致Innodb无法完成初始化，数据库无法启动，但另两个参数可以修改

#innodb_undo_tablespaces = 0

62. InnoDB存储引擎在刷新一个脏页时，会检测该页所在区(extent)的所有页，如果是脏页，那么一刷新。这样做的好处是通过AIO可以将多个IO写操作合并为一个IO操作。对于传统机械硬盘建议使用而对于固态硬盘可以关闭。

#innodb_flush_neighbors = 1

63. 这个值定义了日志文件的大小，innodb日志文件的作用是用来保存redo日志。一个事务对于数据索引的修改往往对应到表空间中的随机的位置，因此当刷新这些修改到磁盘中就会引起随机的I/O，随机的I/O往往比顺序的I/O更加昂贵的开销，因为随机的I/O需要更多的开销来定位到指定的位置。inodb使用日志来将随机的I/O转为顺序的I/O，只要日志文件是安全的，那么事务就是永久的，尽管这改变还没有写到数据文件中，如果出现了当机或服务器断电的情况，那么innodb也可以通过日志文来恢复以及提交的事务。但是日志文件是有一定的大小的，所以必须要把日志文件记录的改变写到数文件中，innodb对于日志文件的操作是循环的，即当日志文件写满后，会将指针重新移动到文件开的地方重新写，但是它不会覆盖那些还没有写到数据文件中的日志，因为这是唯一记录了事务持久化记录

如果对 Innodb 数据表有大量的写入操作，那么选择合适的 innodb_log_file_size 值对提升MySQL能很重要。然而设置太大了，就会增加恢复的时间，因此在MySQL崩溃或者突然断电等情况会令MySL服务器花很长时间来恢复

#innodb_log_file_size = 4G

64. 事务在内存中的缓冲。 分配原 则：控制在2-8M.这个值不用太多的。他里面的内存一般一秒钟写磁盘一次

#innodb_log_buffer_size = 16M

65. 控制是否使用，使用几个独立purge线程（清除二进制日志）

#innodb_purge_threads = 4

66. mysql在5.6之前一直都是单列索引限制767，起因是256×3-1。这个3是字符最大占用空间（utf8。但是在5.6以后，开始支持4个字节的uutf8。255×4>767, 于是增加了这个参数。这个参数默认值OFF。当改为ON时，允许列索引最大达到3072

innodb_large_prefix = 1

67. InnoDB kernel并发最大的线程数。 1) 最少设置为(num_disks+num_cpus)*2。 2) 可以通过设成1000来禁止这个限制

innodb_thread_concurrency = 64

66. 是否将死锁相关信息保存到MySQL 错误日志中

#innodb_print_all_deadlocks = 1

67. 开启InnoDB严格检查模式，尤其采用了页数据压缩功能后，最好是开启该功能。开启此功能后，创建表（CREATE TABLE）、更改表（ALTER TABLE）和创建索引（CREATE INDEX）语句时，如果法有错误，不会有警告信息，而是直接抛出错误，这样就可直接将问题扼杀在摇篮里

#innodb_strict_mode = 1

68. ORDER BY 或者GROUP BY 操作的buffer缓存大小

innodb_sort_buffer_size = 64M

######semi sync replication settings########

69. 指定mysql的插件目录

#plugin_dir=/data/local/mysql-5.7.19/lib/plugin

70. 指定载入哪些插件

#plugin_load = "rpl_semi_sync_master=semisync_master.so;rpl_semi_sync_slave=semisync_slave.so"

71. 控制主库上是否开启semisync

#loose_rpl_semi_sync_master_enabled = 1

72. 控制备库是否开启semisync

#loose_rpl_semi_sync_slave_enabled = 1

73. 单位毫秒，防止半同步复制在没有收到确认的情况下，发送堵塞。master在超时之前没有收到确，将恢复到异步复制，继续执行半同步没有进行的操作

#loose_rpl_semi_sync_master_timeout = 5000

[mysqld-5.7]

74. 表示转储每个bp instance LRU上最热的page的百分比。通过设置该参数可以减少转储的page数

#innodb_buffer_pool_dump_pct = 40

75. 为了提升扩展性和刷脏效率，在5.7.4版本里引入了多个page cleaner线程。从而达到并行刷脏的果

在该版本中，Page cleaner并未和buffer pool绑定，其模型为一个协调线程 + 多个工作线程，协调程本身也是工作线程。因此如果innodb_page_cleaners设置为8，那么就是一个协调线程，加7个工线程

innodb_page_cleaners = 4

76. 是否开启在线回收（收缩）undo log日志文件，支持动态设置

#innodb_undo_log_truncate = 1

77. 当超过这个阀值（默认是1G），会触发truncate回收（收缩）动作，truncate后空间缩小到10M

#innodb_max_undo_log_size = 2G

78. 控制回收（收缩）undo log的频率。undo log空间在它的回滚段没有得到释放之前不会收缩，要增加释放回滚区间的频率，就得降低设定值

#innodb_purge_rseg_truncate_frequency = 128

79. 这个参数控制了当mysql启动或重启时，mysql在搜寻GTIDs时是如何迭代使用binlog文件的。个选项设置为真，会提升mysql执行恢复的性能。因为这样mysql-server启动和binlog日志清理更快该参数为真时，mysql-server只需打开最老的和最新的这2个binlog文件

#binlog_gtid_simple_recovery=1

80. 在MySQL 5.7.2 新增了 log_timestamps 这个参数，该参数主要是控制 error log、genera log 等等记录日志的显示时间参数。 在 5.7.2 之后改参数为默认 UTC 这样会导致日志中记录的时间比中这边的慢，导致查看日志不方便。修改为 SYSTEM 就能解决问题

log_timestamps=system

81. 这个神奇的参数5.7.6版本引入，用于定义一个记录事务的算法，这个算法使用hash标识来记录事。如果使用MGR，那么这个hash值需要用于分布式冲突检测何处理，在64位的系统，官网建议设置参数使用 XXHASH64 算法。如果线上并没有使用该功能，应该设为off

#transaction_write_set_extraction=MURMUR32

82. 从mysql5.7.6开始information_schema.global_status已经开始被舍弃，为了兼容性，此时需要开 show_compatibility_56

#show_compatibility_56=on