

本文介绍了 MySQL 的 InnoDB 和 MyISAM 存储引擎下相关 my.cnf 配置文件配置信息各参数的意义, 对于 MySQL 使用者来讲如果熟悉 my.cnf 配置文件那么就可以更好地理解 MySQL 的运行机制和原理。

basedir = path	使用给定目录作为根目录(安装目录)。
character-sets-dir = path	给出存放着字符集的目录。
datadir = path	从给定目录读取数据库文件。
pid-file = filename	为 mysqld 程序指定一个存放进程 ID 的文件(仅适用于 UNIX/Linux 系统); Init-V 脚本需要使用这个文件里的进程 ID 结束 mysqld 进程。
socket = filename	为 MySQL 客户程序与服务器之间的本地通信指定一个套接字文件(仅适用于 UNIX/Linux 系统; 默认设置一般是/var/lib/mysql/mysql.sock 文件)。在 Windows 环境下, 如果 MySQL 客户与服务器是通过命名管道进行通信的, -sock 选项给出的将是该命名管道的名字(默认设置是 MySQL)。
lower_case_table_name = 1/0	新目录和数据表的名字是否只允许使用小写字母; 这个选项在 Windows 环境下的默认设置是 1(只允许使用小写字母)。

mysqld 程序: 语言设置

character-sets-server = name	新数据库或数据表的默认字符集。为了与 MySQL 的早期版本保持兼容, 这个字符集也可以用 -default-character-set 选项给出; 但这个选项已经显得有点过时了。
collation-server = name	新数据库或数据表的默认排序方式。
language = name	用指定的语言显示出错信息。

mysqld 程序: 通信、网络、信息安全

enable-named-pipes	允许 Windows 2000/XP 环境下的客户和服务使用命名管道(named pipe)进行通信。这个命名管道的默认名字是 MySQL, 但可以用 -socket 选项来改变。
local-infile [=0]	允许/禁止使用 LOAD DATA LOCAL 语句来处理本地文件。
myisam-recover [=opt1, opt2, ...]	在启动时自动修复所有受损的 MyISAM 数据表。这个选项的可取值有 4 种: DEFAULT、BACKUP、QUICK 和 FORCE; 它们与 myisamchk 程序的同名选项作用相同。
old-passwords	使用 MySQL 3.23 和 4.0 版本中的老算法来加密 mysql 数据库里的密码(默认使用 MySQL 4.1

	版本开始引入的新加密算法)。
port = n	为 MySQL 程序指定一个 TCP/IP 通信端口(通常是 3306 端口)。
safe-user-create	只有在 mysql.user 数据库表上拥有 INSERT 权限的用户才能使用 GRANT 命令；这是一种双保险机制(此用户还必须具备 GRANT 权限才能执行 GRANT 命令)。
shared-memory	允许使用内存(shared memory)进行通信(仅适用于 Windows)。
shared-memory-base-name = name	给共享内存块起一个名字(默认的名字是 MySQL)。
skip-grant-tables	不使用 mysql 数据库里的信息来进行访问控制(警告:这将允许用户任何用户去修改任何数据库)。
skip-host-cache	不使用高速缓存区来存放主机名和 IP 地址的对应关系。
skip-name-resolve	不把 IP 地址解析为主机名；与访问控制(mysql.user 数据表)有关的检查全部通过 IP 地址行进。
skip-networking	只允许通过一个套接字文件(Unix/Linux 系统)或通过命名管道(Windows 系统)进行本地连接，不允许 TCP/IP 连接；这提高了安全性，但阻断了来自网络的外部连接和所有的 Java 客户程序(Java 客户即使在本地连接里也使用 TCP/IP)。
user = name	mysqld 程序在启动后将在给定 UNIX/Linux 账户下执行；mysqld 必须从 root 账户启动才能在启动后切换到另一个账户下执行；mysqld_safe 脚本将默认使用 -user=mysql 选项来启动 mysqld 程序。

mysqld 程序：内存管理、优化、查询缓存区

bulk_insert_buffer_size = n	为一次插入多条新记录的 INSERT 命令分配的缓存区长度(默认设置是 8M)。
key_buffer_size = n	用来存放索引区块的 RMA 值(默认设置是 8M)。
join_buffer_size = n	在参加 JOIN 操作的数据列没有索引时为 JOIN 操作分配的缓存区长度(默认设置是 128K)。
max_heap_table_size = n	HEAP 数据表的最大长度(默认设置是 16M)；超过这个长度的 HEAP 数据表将被存入一个临时文件而不是驻留在内存里。
max_connections = n	MySQL 服务器同时处理的数据库连接的最大数量(默认设置是 100)。

query_cache_limit = n	允许临时存放在查询缓存区里的查询结果的最大长度(默认设置是 1M)。
query_cache_size = n	查询缓存区的最大长度(默认设置是 0，不开辟查询缓存区)。
query_cache_type = 0/1/2	查询缓存区的工作模式:0, 禁用查询缓存区; 1, 启用查询缓存区(默认设置); 2, "按需分配"模式, 只响应 SELECT SQL_CACHE 命令。
read_buffer_size = n	为从数据表顺序读取数据的读操作保留的缓存区的长度(默认设置是 128KB); 这个选项的设置值在必要时可以用 SQL 命令 SET SESSION read_buffer_size = n 命令加以改变。
read_rnd_buffer_size = n	类似于 read_buffer_size 选项, 但针对的是按某种特定顺序(比如使用了 ORDER BY 子句的查询)输出的查询结果(默认设置是 256K)。
sort_buffer = n	为排序操作分配的缓存区的长度(默认设置是 2M); 如果这个缓存区太小, 则必须创建一个临时文件来进行排序。
table_cache = n	同时打开的数据表的数量(默认设置是 64)。
tmp_table_size = n	临时 HEAP 数据表的最大长度(默认设置是 32M); 超过这个长度的临时数据表将被转换为 MyISAM 数据表并存入一个临时文件。

mysqld 程序: 日志

log [= file]	把所有的连接以及所有的 SQL 命令记入日志(通用查询日志); 如果没有给出 file 参数, MySQL 将在数据库目录里创建一个 hostname.log 文件作为这种日志文件(hostname 是服务器的主机名)。
log-slow-queries [= file]	把执行用时超过 long_query_time 变量值的查询命令记入日志(慢查询日志); 如果没有给出 file 参数, MySQL 将在数据库目录里创建一个 hostname-slow.log 文件作为这种日志文件(hostname 是服务器主机名)。
long_query_time = n	慢查询的执行用时上限(默认设置是 10s)。
long_queries_not_using_indexes	把慢查询以及执行时没有使用索引的查询命令全都记入日志(其余同 -log-slow-queries 选项)。
log-bin [= filename]	把对数据进行修改的所有 SQL 命令(也就是 INSERT、UPDATE 和 DELETE 命令)以二进制格式记入日志(二进制变更日志, binary update log)。这种日志的文件名是 filename.n 或默认的 hostname.n, 其中 n 是一个 6 位数字的整数(日志文件按顺序编号)。

log-bin-index = filename	二进制日志功能的索引文件名。在默认情况下，这个索引文件与二进制日志文件的名字相同，但后缀名是 .index 而不是 .nnnnnn。
max_binlog_size = n	二进制日志文件的最大长度(默认设置是 1GB)。在前一个二进制日志文件里的信息量超过这个最大长度之前，MySQL 服务器会自动提供一个新的二进制日志文件接续上。
binlog-do-db = dbname	只把给定数据库里的变化情况记入二进制日志文件，其他数据库里的变化情况不记载。如果需要记载多个数据库里的变化情况，就必须在配置文件使用多个本选项来设置，每个数据库一行。
binlog-ignore-db = dbname	不把给定数据库里的变化情况记入二进制日志文件。
sync_binlog = n	每经过 n 次日志写操作就把日志文件写入硬盘一次(对日志信息进行一次同步)。n=1 是最安全的做法，但效率最低。默认设置是 n=0，意思是由操作系统来负责二进制日志文件的同步工作。
log-update [= file]	记载出错情况的日志文件名(出错日志)。这种日志功能无法禁用。如果没有给出 file 参数，MySQL 会使用 hostname.err 作为种日志文件的名字。

mysqld 程序：镜像(主控镜像服务器)

server-id = n	给服务器分配一个独一无二的 ID 编号；n 的取值范围是 1~2 的 32 次方启用二进制日志功能。
log-bin = name	启用二进制日志功能。这种日志的文件名是 filename.n 或默认的 hostname.n，其中的 n 是一个 6 位数字的整数(日志文件顺序编号)。
binlog-do/ignore-db = dbname	只把给定数据库里的变化情况记入二进制日志文件/不把给定的数据库里的变化记入二进制日志文件。

mysqld 程序：镜像(从属镜像服务器)

server-id = n	给服务器分配一个唯一的 ID 编号
log-slave-updates	启用从属服务器上的日志功能，使这台计算机可以用来构成一个镜像链(A->B->C)。
master-host = hostname	主控服务器的主机名或 IP 地址。如果从属服务器上存在 mater.info 文件(镜像关系定义文件)，它将忽略此选项。
master-user = replicusername	从属服务器用来连接主控服务器的用户名。如果从属服务器上存在 mater.info 文件，它将忽略此选项。

master-password = passwd	从属服务器用来连接主控服务器的密码。如果从属服务器上存在 mater.info 文件，它将忽略此选项。
master-port = n	从属服务器用来连接主控服务器的 TCP/IP 端口 (默认设置是 3306 端口)。
master-connect-retry = n	如果与主控服务器的连接没有成功，则等待 n 秒(s)后再进行管理方式 (默认设置是 60s)。如果从属服务器存在 mater.info 文件，它将忽略此选项。
master-ssl-xxx = xxx	对主、从服务器之间的 SSL 通信进行配置。
read-only = 0/1	0: 允许从属服务器独立地执行 SQL 命令 (默认设置); 1: 从属服务器只能执行来自主控服务器的 SQL 命令。
read-log-purge = 0/1	1: 把处理完的 SQL 命令立刻从中继日志文件里删除 (默认设置); 0: 不把处理完的 SQL 命令立刻从中继日志文件里删除。
replicate-do-table = dbname. tablename	与 - replicate-do-table 选项的含义和用法相同，但数据库和数据库表名字里允许出现通配符“%” (例如: test%.%- 对名字以“test” 开头的所有数据库里的所有数据库表进行镜像处理)。
replicate-do-db = name	只对这个数据库进行镜像处理。
replicate-ignore-table = dbname. tablename	不对这个数据表进行镜像处理。
replicate-wild-ignore-table = dbn. tablen	不对这些数据表进行镜像处理。
replicate-ignore-db = dbname	不对这个数据库进行镜像处理。
replicate-rewrite-db = db1name > db2name	把主控数据库上的 db1name 数据库镜像处理为从属服务器上的 db2name 数据库。
report-host = hostname	从属服务器的主机名; 这项信息只与 SHOW SLAVE HOSTS 命令有关 - 主控服务器可以用这条命令生成一份从属服务器的名单。
slave-compressed-protocol = 1	主、从服务器使用压缩格式进行通信 - 如果它们都支持这么做的话。

slave-skip-errors = n1, n2, ... 或 all	即使发生出错代码为 n1、n2 等的错误，镜像处理工作也继续进行(即不管发生什么错误，镜像处理工作也继续进行)。如果配置得当，从属服务器不应该在执行 SQL 命令时发生错误(在 主控服务器上执行出错的 SQL 命令不会被发送到从属服务器上做镜像处理)；如果不使用 slave-skip-errors 选项，从属服务器上的镜像工作就可能因为发生错误而中断，中断后需要有人工参与才能继续进行。
------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

mysqld - InnoDB: 基本设置、表空间文件

skip-innodb	不加载 InnoDB 数据表驱动程序 - 如果用不着 InnoDB 数据表，可以用这个选项节省一些内存。
innodb-file-per-table	为每一个新数据表创建一个表空间文件而不是把数据表都集中保存在中央表空间里(后者是默认设置)。该选项始见于 MySQL 4.1。
innodb-open-file = n	InnoDB 数据表驱动程序最多可以同时打开的文件数(默认设置是 300)。如果使用了 innodb-file-per-table 选项并且需要同时打开很多数据表的话，这个数字很可能需要加大。
innodb_data_home_dir = p	InnoDB 主目录，所有与 InnoDB 数据表有关的目录或文件路径都相对于这个路径。在默认的情况下，这个主目录就是 MySQL 的数据目录。
innodb_data_file_path = ts	用来容纳 InnoDB 为数据表的表空间：可能涉及一个以上的文件，每一个表空间文件的最大长度都必须以字节(B)、兆字节(MB)或千兆字节(GB)为单位给出；表空间文件的名字必须以分号隔开；最后一个表空间文件还可以带一个 autoextend 属性和一个最大长度(max:n)。例如，ibdata1:1G; ibdata2:1G:autoextend:max:2G 的意思是：表空间文件 ibdata1 的最大长度是 1GB，ibdata2 的最大长度也是 1G，但允许它扩充到 2GB。除文件名外，还可以用硬盘分区的设置名来定义表空间，此时必须给表空间的最大初始长度值加上 newraw 关键字做后缀，给表空间的最大扩充长度值加上 raw 关键字做后缀(例如/dev/hdb1: 20Gnewraw 或/dev/hdb1:20Graw)；MySQL 4.0 及更高版本的默认设置是 ibdata1:10M:autoextend。
innodb_autoextend_increment = n	带有 autoextend 属性的表空间文件每次加大多少兆字节(默认设置是 8MB)。这个属性不涉及具体的数据表文件，那些文件的增大速度相对是比较小的。
innodb_lock_wait_timeout = n	如果某个事务在等待 n 秒(s)后还没有获得所需要的资源，就使用 ROLLBACK 命令放弃这个事务。这项设置对于发现和处理未能被 InnoDB 数据表驱动程序识别出来的死锁条件有着重要的意义。这个选项的默认设置是 50s。
innodb_fast_shutdown 0/1	是否以最快的速度关闭 InnoDB，默认设置是 1，意思是不把缓存在 INSERT 缓存区的数据写入数据表，那些数据将在 MySQL 服务器下次启动时再写入(这么做没有什么风险，因为 INSERT 缓

	存区是表空间的一个组成部分, 数据不会丢失)。把这个选项设置为 0 反而危险, 因为在计算机关闭时, InnoDB 驱动程序很可能没有足够的时间完成它的数据同步工作, 操作系统也许会在它完成数据同步工作之前强行结束 InnoDB, 而这会导致数据不完整。
--	---------------------------------------------------------------------------------------------------------------------------------

mysqld 程序: InnoDB - 日志

innodb_log_group_home_dir = p	用来存放 InnoDB 日志文件的目录路径(如 ib_logfile0、ib_logfile1 等)。在默认的情况下, InnoDB 驱动程序将使用 MySQL 数据目录作为自己保存日志文件的位置。
innodb_log_files_in_group = n	使用多少个日志文件(默认设置是 2)。InnoDB 数据表驱动程序将以轮转方式依次填写这些文件: 当所有的日志文件都写满以后, 之后的日志信息将写入第一个日志文件的最大长度(默认设置是 5MB)。这个长度必须以 MB(兆字节)或 GB(千兆字节)为单位进行设置。
innodb_flush_log_at_trx_commit = 0/1/2	这个选项决定着什么时候把日志信息写入日志文件以及什么时候把这些文件物理地写(术语称为“同步”)到硬盘上。设置值 0 的意思是每隔一秒写一次日志并进行同步, 这可以减少硬盘写操作次数, 但可能造成数据丢失; 设置值 1(设置设置)的意思是在每执行完一条 COMMIT 命令就写一次日志并进行同步, 这可以防止数据丢失, 但硬盘写操作可能会很频繁; 设置值 2 是一般折衷的办法, 即每执行完一条 COMMIT 命令写一次日志, 每隔一秒进行一次同步。
innodb_flush_method = x	InnoDB 日志文件的同步办法(仅适用于 UNIX/Linux 系统)。这个选项的可取值有两种: fdatasync, 用 fsync() 函数进行同步; O_DSYNC, 用 O_SYNC() 函数进行同步。
innodb_log_archive = 1	启用 InnoDB 驱动程序的 archive(档案)日志功能, 把日志信息写入 ib_arch_log_n 文件。启用这种日志功能在 InnoDB 与 MySQL 一起使用时没有多大意义(启用 MySQL 服务器的二进制日志功能就足够用了)。

mysqld 程序 - InnoDB: 缓存区的设置和优化

innodb_log_buffer_pool_size = n	为 InnoDB 数据表及其索引而保留的 RAM 内存量(默认设置是 8MB)。这个参数对速度有着相当大的影响, 如果计算机上只运行有 MySQL/InnoDB 数据库服务器, 就应该把全部内存的 80%用于这个用途。
innodb_log_buffer_size = n	事务日志文件写操作缓存区的最大长度(默认设置是 1MB)。
innodb_additional_mem_pool_size =	为用于内部管理的各种数据结构分配的缓存区最大长度(默认设置是 1MB)。

n	
innodb_file_io_threads = n	I/O 操作(硬盘写操作)的最大线程个数(默认设置是 4)。
innodb_thread_concurrency = n	InnoDB 驱动程序能够同时使用的最大线程个数(默认设置是 8)。

mysqld 程序：其它选项

bind-address = ipaddr	MySQL 服务器的 IP 地址。如果 MySQL 服务器所在的计算机有多个 IP 地址，这个选项将非常重要。
default-storage-engine = type	新数据表的默认数据表类型(默认设置是 MyISAM)。这项设置还可以通过 <code>--default-table-type</code> 选项来设置。
default-timezone = name	为 MySQL 服务器设置一个地理时区(如果它与本地计算机的地理时区不一样)。
ft_min_word_len = n	全文索引的最小单词长度。这个选项的默认设置是 4，意思是在创建全文索引时不考虑那些由 3 个或更少的字符构建单词。
Max-allowed-packet = n	客户与服务器之间交换的数据包的最大长度，这个数字至少应该大于客户程序将要处理的最大 BLOB 块的长度。这个选项的默认设置是 1MB。
Sql-mode = mode1, mode2, ...	MySQL 将运行在哪一种 SQL 模式下。这个选项的作用是让 MySQL 与其他的数据库系统保持最大程度的兼容。这个选项的可取值包括 <code>ansi</code> 、 <code>db2</code> 、 <code>oracle</code> 、 <code>no_zero_date</code> 、 <code>pipes_as_concat</code> 。