my.cnf具体的配置说明以下：算法

|  |  |
| --- | --- |
| basedir = path | 使用给定目录做为根目录(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脚本须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讯指定一个套接字文件(仅适用于UNIX/Linux系统; 默认设置通常是/var/lib/mysql/mysql.sock文件)。在Windows环境下，若是MySQL客户与服务器是经过命名管道进行通讯 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容许使用小写字母; 这个选项在Windows环境下的默认设置是1(只容许使用小写字母)。 |

**mysqld程序：语言设置**sql

|  |  |
| --- | --- |
| 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，这个字符集也能够用–default-character-set选项给出; 但这个选项已经显得有点过期了。 |
| collation-server = name | 新数据库或数据表的默认排序方式。 |
| lanuage = name | 用指定的语言显示出错信息。 |

**mysqld程序：通讯、网络、信息安全**数据库

|  |  |
| --- | --- |
| enable-named-pipes | 容许Windows 2000/XP环境下的客户和服务器使用命名管道(named pipe)进行通讯。这个命名管道的默认名字是MySQL，但能够用–socket选项来改变。 |
| local-infile [=0] | 容许/禁止使用LOAD DATA LOCAL语句来处理本地文件。 |
| myisam-recover [=opt1, opt2, ...] | 在启动时自动修复全部受损的MyISAM数据表。这个选项的可取值有4种:DEFAULT、BACKUP、QUICK和FORCE; 它们与myisamchk程序的同名选项做用相同。 |
| 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通讯端口(一般是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(此用户还必须具有GRANT权限才能执行GRANT命令)。 |
| shared-memory | 容许使用内存(shared memory)进行通讯(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:这将容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-resovle | 不把IP地址解析为主机名; 与访问控制(mysql.user数据表)有关的检查所有经过IP地址行进。 |
| skip-networking | 只容许经过一个套接字文件(Unix/Linux系统)或经过命名管道(Windows系统)进行本地链接，不容许ICP/IP链接; 这提升了安全性，但阻断了来自网络的外部链接和全部的Java客户程序(Java客户即便在本地链接里也使用TCP/IP)。 |
| user = name | mysqld程序在启动后将在给定UNIX/Linux帐户下执行; mysqld必须从root帐户启动才能在启动后切换到另外一个帐户下执行; 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**缓存

|  |  |
| --- | --- |
| 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操做的数据列没有索引时为JOIN操做分配的缓存区长度(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链接的最大数量(默认设置是100)。 |
| query\_cache\_limit = n | 容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 | 查询缓存区的工做模式: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做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能够用SQL命令SET SESSION read\_buffer\_size = n命令加以改变。 |
| read\_rnd\_buffer\_size = n | 相似于read\_buffer\_size选项，但针对的是按某种特定顺序(好比使用了ORDER BY子句的查询)输出的查询结果(默认设置是256K)。 |
| sore\_buffer = n | 为排序操做分配的缓存区的长度(默认设置是2M); 若是这个缓存区过小，则必须建立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**安全

|  |  |
| --- | --- |
| log [= file] | 把全部的链接以及全部的SQL命令记入日志(通用查询日志); 若是没有给出file参数，MySQL将在数据库目录里建立一个hostname.log文件做为这种日志文件(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); 若是没有给出file参数，MySQL将在数据库目录里建立一个hostname-slow.log文件做为这种日志文件(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\_indexs | 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(其他同–log-slow-queries选项)。 |
| log-bin [= filename] | 把对数据进行修改的全部SQL命令(也就是INSERT、UPDATE和DELETE命令)以二进制格式记入日志(二进制变动日志，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。在默认状况下，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以前，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 据库里的变化状况记入二进制日志文件，其余数据库里的变化状况不记载。若是须要记载多个数据库里的变化状况，就必须在配置文件使用多个本选项来设置，每一个数据库一行。 |
| binlog-ignore-db = dbname | 不把给定数据库里的变化状况记入二进制日志文件。 |
| sync\_binlog = n | 每通过n第二天志写操做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是最安全的作法，但效率最低。默认设置是n=0，意思是由操做系统来负责二进制日志文件的同步工做。 |
| log-update [= file] | 记载出错状况的日志文件名(出错日志)。这种日志功能没法禁用。若是没有给出file参数，MySQL会使用hostname.err做为种日志文件的名字。 |

**mysqld程序：镜像(主控镜像服务器)**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状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)**网络

|  |  |
| --- | --- |
| server-id = n | 给服务器分配一个惟一的ID编号 |
| log-slave-updates | 启用从属服务器上的日志功能，使这台计算机能够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。若是从属服务器上存在mater.info文件(镜像关系定义文件)，它将忽略此选项。 |
| master-user = replicusername | 从属服务器用来链接主控服务器的用户名。若是从属服务器上存在mater.info文件，它将忽略此选项。 |
| master-password = passwd | 从属服务器用来链接主控服务器的密码。若是从属服务器上存在mater.info文件，它将忽略此选项。 |
| master-port = n | 从属服务器用来链接主控服务器的TCP/IP端口(默认设置是3306端口)。 |
| master-connect-retry = n | 若是与主控服务器的链接没有成功，则等待n秒(s)后再进行管理方式(默认设置是60s)。若是从属服务器存在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通讯进行配置。 |
| read-only = 0/1 | 0: 容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命令马上从中继日志文件里删除(默认设置); 0: 不把处理完的SQL命令马上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容许出现通配符”%” (例如: test%.%–对名字以”test”开头的全部数据库里的因此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–主控服务器能够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讯–若是它们都支持这么作的话。 |
| slave-skip-errors = n1, n2, …或all | 即便发生出错代码为n一、n2等的错误，镜像处理工做也继续进行(即无论发生什么错误，镜像处理工做也继续进行)。若是配置得当，从属服务器不该 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作镜像处理); 若是不使用slave-skip-errors选项，从属服务器上的镜像工做就可能由于发生错误而中断，中断后须要有人工参与才能继续进行。 |

**mysqld–InnoDB：基本设置、表空间文件**数据结构

|  |  |
| --- | --- |
| skip-innodb | 不加载InnoDB数据表驱动程序–若是用不着InnoDB数据表，能够用这个选项节省一些内存。 |
| innodb-file-per-table | 为每个新数据表建立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能够同时打开的文件数(默认设置是300)。若是使用了innodb-file-per-table选项而且须要同时打开不少数据表的话，这个数字极可能须要加大。 |
| innodb\_data\_home\_dir = p | InnoDB主目录，全部与InnoDB数据表有关的目录或文件路径都相对于这个路径。在默认的状况下，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能够带一个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，但容许它扩充到2GB。除文件名外，还能够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关键字作后缀，给表空间的最大扩充长度值加上raw关键字作后缀(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 | 若是某个事务在等待n秒(s)后尚未得到所须要的资源，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 (这么作没有什么风险，由于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反面危险，由于在计算机关闭时，InnoDB 驱动程序极可能没有足够的时间完成它的数据同步工做，操做系统也许会在它完成数据同步工做以前强行结束InnoDB，而这会致使数据不完整。 |

**mysqld程序：InnoDB–日志**oracle

|  |  |
| --- | --- |
| 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)。在默认的状况下，InnoDB驱动程序将使用 MySQL数据目录做为本身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; 当全部的日志文件都写满之后，以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 | 这个选项决定着何时把日志信息写入日志文件以及何时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第二天 志并进行 同步，这能够减小硬盘写操做次数，但可能形成数据丢失; 设置值1(设置设置)的意思是在每执行完一条COMMIT命令就写一第二天志并进行同步，这能够防止数据丢失，但硬盘写操做可能会很频繁; 设置值2是通常折衷的办法，即每执行完一条COMMIT命令写一第二天志，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一块儿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至关大的影响，若是计算机上只运行有 MySQL/InnoDB数据库服务器，就应该把所有内存的80%用于这个用途。 |
| innodb\_log\_buffer\_size = n | 事务日志文件写操做缓存区的最大长度(默认设置是1MB)。 |
| innodb\_additional\_men\_pool\_size = n | 为用于内部管理的各类数据结构分配的缓存区最大长度(默认设置是1MB)。 |
| innodb\_file\_io\_threads = n | I/O操做(硬盘写操做)的最大线程个数(默认设置是4)。 |
| innodb\_thread\_concurrency = n | InnoDB驱动程序可以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。若是MySQL服务器所在的计算机有多个IP地址，这个选项将很是重要。 |
| default-storage-engine = type | 新数据表的默认数据表类型(默认设置是MyISAM)。这项设置还能够经过–default-table-type选项来设置。 |
| default-timezone = name | 为MySQL服务器设置一个地理时区(若是它与本地计算机的地理时区不同)。 |
| ft\_min\_word\_len = n | 全文索引的最小单词长度工。这个选项的默认设置是4，意思是在建立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 = model1, mode2, … | MySQL将运行在哪种SQL模式下。这个选项的做用是让MySQL与其余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ansi、db二、 oracle、no\_zero\_date、pipes\_as\_concat。 |

注意：若是在配置文件里给出的某个选项是mysqld没法识别的，MySQL服务器将不启动。