

MYSQL-innodb 性能优化学习总结

BSS 测试部: newhackerman

数据库参数

MYSQL 数据库的参数配置一般在 my.ini 配置文件中修改/添加（部分参数也可以用 `set global 参数名=值` 做临时调整，重启后失效），配置完后需要重启数据库才生效。

参数 1: `innodb_buffer_pool_size = Gb/MB`

说明：此参数类似于 oracle 的 SGA 配置，当主机做为 mysql 数据库服务器时，一般配置为整机内存的 60%~80%。

参数 2: `innodb_buffer_pool_instances=N`

说明：内存缓冲池实例数，将 `innodb_buffer_pool_size` 配置的内存分割成 N 份，此参数当配置内存大小小于 1G 时才生效，当数据库有多个会话进行数据库操作时，用于并行在多个内存块中处理任务，一般配置值《=服务器 CPU 的个数。

参数 3: `max_connections = 2000`

说明：最大连接数，当数据库面对高并发时，这个值需要调节为一个合理的值，才满足业务的并发要求，避免数据库拒绝连接。

参数 4: `max_user_connections=1000`

说明：设置单个用户的连接数。

参数 5: `innodb_log_buffer_size =32M`

说明：日志缓冲区大小，一般不用设置太大，能存下 1 秒钟操作的数据日志就行了，mysql 默认 1 秒写一轮询写一次日志到磁盘。

参数 6: `innodb_flush_log_at_trx_commit`

说明：（这个配置很关键）一般的实时业务交易配置为 2，取值 0,1,2

0:数据操作时，直接写内存，并不同时写入磁盘；

2:数据操作时，直接写内存，并不同时写入磁盘；

1:就每个事务提交就会要刷新到磁盘后才算提交完成,这种情况是保证了事务的一致性,但性能会有很大的影响。

0 与 2 的区别：

0：当 mysql 挂了之后，可能会损失前一秒的事务信息

2：当 mysql 挂了之后，如果系统文件系统没挂，不会有事务丢失。

参数 7: `innodb_read_io_threads = 16`

说明：数据库读操作时的线程数，用于并发。

参数 8: `innodb_write_io_threads = 16`

说明：数据库写操作时的线程数，用于并发。

参数 9: innodb file per table= 1

说明：每一个表是否使用独立的数据表空间，默认为 OFF（使用共享表空间），一般建议配置为 1，InnoDB 默认会将所有的数据库 InnoDB 引擎的表数据存储在一个共享空间中：（ibdata1），这样就感觉不爽，增删数据库的时候，ibdata1 文件不会自动收缩，单个数据库的备份也将成为问题。通常只能将数据使用 mysqldump 导出，然后再导入解决这个问题。

共享表空间在 Insert 操作上少有优势。其它都没独立表空间表现好，如果数据库基本上都插入操作则配置为 0。

参数 10: innodb_stats_on_metadata={ OFF|on}

说明：是否动态收集统计信息，开启时会影响数据库的性能（一般关闭，找个时间手动刷新，或定时刷新）如果为关闭时，需要配置数据库调度任务，定时刷新数据库的统计信息。

参数 11: innodb_spin_wait_delay=10

说明：控制 CPU 的轮询时间间隔，默认是 6，配置过低时，任务调度比较频繁，会消耗 CPU 资源。

参数 12: innodb_lock_wait_timeout=30

说明：控制锁的超时时间，默认为 50，这个值要注意，如果有特殊业务确实要耗时较长时，不能配置太短。

执行计划分析

explain sql语句：

```
mysql> explain select * from test; 或: explain select * from test \G;
```

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	test	ALL	NULL	NULL	NULL	NULL	82	NULL

—字段解释：

输入 explain select * from customer; 后，出现一张表，个行的意思如下：

table—显示此行数据属于哪张表；

type—重要的一列，显示使用了何种连接，从好到差依次为const、eq_ref、ref、

range、index、all，下面详细说明：

type的描述：

system—表只有一行，这是const连接类型的特例；

const—表中一个记录的最大值能够匹配这个查询（索引可以是主键或唯一索引）。因

为只有一行，这个值实际就是常数，因为mysql先读这个值，再把它当作常数对待

eq_ref—从前面的表中，对每一个记录的联合都从表中读取一个记录。在查询使用索引为主键或唯一索引的全部时使用；

ref—只有使用了不是主键或唯一索引的部分时发生。对于前面表的每一行联合，全部记录都将从表中读出，这个连接类型严重依赖索引匹配记录的多少—越少越好；

range—使用索引返回一个范围中的行，比如使用>或<查找时发生；

index—这个连接类型对前面的表中的每一个记录联合进行完全扫描(比all好，因为索引一般小于表数据)；

all—这个连接类型多前面的表中的每一个记录联合进行完全扫描，这个比较糟糕，应该尽量避免。

possible_keys—可以应用在这张表中的索引，如果为null，则表示没有可用索引；

key—实际使用的索引，如为null，表示没有用到索引；

key_len—索引的长度，在不损失精确度的情况下，越短越好；

ref—显示索引的哪一列被使用了，如果可能的话，是个常数；

rows—返回请求数据的行数；

extra—关于mysql如何解析查询的额外信息，下面会详细说明。

extra行的描述：

distinct—mysql找到了域行联合匹配的行，就不再搜索了；

not exists—mysql优化了left join，一旦找到了匹配left join的行，就不再搜索了；

range checked for each—没找到理想的索引，一次对于从前面表中来的每一个行组合；

record(index map: #)—检查使用哪个索引，并用它从表中返回行，这是使用索引最慢的一种；

using filesort—看到这个就需要优化查询了，mysql需要额外的步骤来发现如何对返回的行排序。他根据连接类型以及存储排序键值和匹配条件的全部行的行指针来排序全部行。

using index—列数据是从单使用了索引中的信息而没有读取实际行的表返回的，这发生在对表的全部的请求列都是同一个索引时；

using temporary一看到这个就需要优化查询了，mysql需要创建一个临时表来查询存储结果，这通常发生在多不同的列表进行order by时，而不是group by；

where used一使用了where从句来限制哪些行将与下一张表匹配或是返回给用户。如不想返回表中用的全部行，并连接类型是all或index，这就会发生，也可能是查询有问题。

SQL 语句优化：注意 SQL 语句的书写规则，where 条件，order by ,group by ,having ,in ,like ,join on, 表顺序，聚合函数的使用，子查询等。

索引优化

1. 是否有无重复索引
2. 索引字段类型，顺序是否合理
3. 是否有无用索引
4. 索引利用率

表结构优化

1. 表的字段类型是否合理
2. 数据是否冗余
3. 根据业务规则建立合理的约束
4. 建表时，尽量使字段值不为空(加not null约束)，索引列值尽量离散（不重复）
5. 不常用的字段列可适当考虑拆分表
6. 数据量较大的表，有存储时间，IP地址数据时，转为int ,bigint

INT类型的时间数据转换：

UNIX_TIMESTAMP('2015-01-10 12:00:00') 转int（插入数据时）

FROM_UNIXTIME(时间字段) 取时间字段的值

IP地址数据操作转换：

INET_ATON :IP地址转bigint (inet_aton(192.168.1.1))

INET_NTOA :BIGINT转IP地址 (inet_ntoa(ip地址字段))

在mysql中int ,比varchar处理要简单，尽量少使用text类型

数据库主机参数优化

网络：（/etc/security/sysctl.conf）

net.ipv4.tcp_keepalive_time = 1200

说明：表示当keepalive起用的时候，TCP发送keepalive消息的频度。缺省是2小时，改为20分钟。

net.ipv4.ip_local_port_range = 10000 65000

说明：表示用于向外连接的端口范围，一般低位端口不要设置太低，有可能会用到其它程序固定的端口

net.ipv4.tcp_max_syn_backlog = 65535

说明：表示SYN队列的长度，默认为1024，加大队列长度为65535，可以容纳更多等待连接的网络连接数。

net.ipv4.tcp_max_tw_buckets = 5000

说明：示系统同时保持TIME_WAIT的最大数量，如果超过这个数字，TIME_WAIT将立刻被清除并打印警告信息。默认为180000，避免被大量的timewait拖死。

net.ipv4.tcp_syncookies = 1

说明：表示开启SYN Cookies

net.ipv4.tcp_tw_reuse=1

说明：表示开启重用，允许将TIME-WAIT sockets重新用于新的TCP连接

net.ipv4.tcp_recycle=1

说明：表示开启TCP连接中TIME-WAIT sockets的快速回收，默认为0，表示关闭；

net.ipv4.tcp_fin_timeout=10

说明：修改系统默认的 TIMEOUT 时间。

limit.conf内核相关参数优化：

详见：《limits.conf 详解》



limits.conf详解.txt

数据库参数调节效果对比

下列参数配置更改前与更改后在笔记本电脑上的测试结果对比。

```
innodb_buffer_pool_size = 1024M
innodb_buffer_pool_instances = 4
max_connections = 2000
max_user_connections = 2000
innodb_flush_log_at_trx_commit = 2
innodb_write_io_threads = 16
innodb_read_io_threads = 16
innodb_spin_wait_delay = 10
innodb_lock_wait_timeout = 30
```

更改前：增查删操作通过 jmeter 测试每秒只有 80.8 笔。

The screenshot shows the JMeter Aggregate Report window. The 'Test Plan' tree on the left includes '用户定义的变量', 'JDBC Connect', 'JDBC Connect', 'Thread Group', '循环控制器', '循环控制器', '聚合报告', 'View Results', and '工作台'. The '聚合报告' (Aggregate Report) window is active, displaying the following data:

Label	# Samples	Av...	Median	90% Line	Min	Max	E...	Throug...	KB/sec
JDBC Request_insert	8137	1229	1192	2224	16	6300	0.05%	27.0/sec	.2
JDBC Request_select	8094	1074	1038	2076	0	6741	0.00%	26.9/sec	2.5
JDBC Request_delete	8069	1372	1285	2296	32	6767	0.00%	26.8/sec	.2
总体	24300	1225	1172	2198	0	6767	0.02%	80.8/sec	3.0

The 'Throug...' column values (27.0/sec, 26.9/sec, 26.8/sec, and 80.8/sec) are circled in red.

更改后：增查删操作通过 jmeter 测试每秒达到 1513.1 笔。

The screenshot shows the JMeter Aggregate Report window after optimization. The 'Test Plan' tree on the left is the same as the previous screenshot. The '聚合报告' (Aggregate Report) window is active, displaying the following data:

Label	# Samples	Av...	Median	90% Line	Min	Max	E...	Throug...	KB/sec
JDBC Request_insert	151364	63	26	177	0	1327	0.03%	504.5/sec	4
JDBC Request_select	151328	58	24	168	0	1146	0.00%	504.4/sec	47
JDBC Request_delete	151294	64	26	183	0	1129	0.00%	504.3/sec	4
总体	453986	62	25	176	0	1327	0.01%	1513.1/sec	56

The 'Throug...' column values (504.5/sec, 504.4/sec, 504.3/sec, and 1513.1/sec) are circled in red.