三个方法优化MySQL数据库查询 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/259/2021_2022__E4_B8_89_E4_B8_AA_E6_96_B9_E6_c98_259362.htm 在优化查询中，数据库应用（如MySQL）即意味着对工具的操作与使用。使用索引、使用EXPLAIN分析查询以及调整MySQL的内部配置可达到优化查询的目的。 任何一位数据库程序员都会有这样的体会：高通信量的数据库驱动程序中，一条糟糕的SQL查询语句可对整个应用程序的运行产生严重的影响，其不仅消耗掉更多的数据库时间，且它将对其他应用组件产生影响。 如同其它学科，优化查询性能很大程度上决定于开发者的直觉。幸运的是，像MySQL这样的数据库自带有一些协助工具。本文简要讨论诸多工具之三种：使用索引，使用EXPLAIN分析查询以及调整MySQL的内部配置。#1: 使用索引 MySQL允许对数据库表进行索引，以此能迅速查找记录，而无需一开始就扫描整个表，由此显着地加快查询速度。每个表最多可以做到16个索引，此外MySQL还支持多列索引及全文检索。 给表添加一个索引非常简单，只需调用一个CREATE INDEX命令并为索引指定它的域即可。列表A给出了一个例子：列表 Amysql> CREATE INDEX idx_username ON users(username).Query OK, 1 row affected (0.15 sec)Records: 1 Duplicates: 0 Warnings: 0 这里，对users表的username域做索引，以确保在WHERE或者HAVING子句中引用这一域的SELECT查询语句运行速度比没有添加索引时要快。通过SHOW INDEX命令可以查看索引已被创建（列表B）。列表 Bmysql> SHOW INDEX FROM users.----------

---------- --------- ----------- --------- ------- ------ ---------- --------- | Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment |------------ ------------ ---------- ----------- --------- ------- ------ ----------- --------- | users | 1 | idx_username | 1 | username | A | NULL | NULL | NULL | YES | BTREE | |------------ ------------ ---------- ----------- --------- ------- ------ ----------- --------- 1 row in set (0.00 sec) 值得注意的是：索引就像一把双刃剑。对表的每一域做索引通常没有必要，且很可能导致运行速度减慢，因为向表中插入或修改数据时，MySQL不得不每次都为这些额外的工作重新建立索引。另一方面，避免对表的每一域做索引同样不是一个非常好的主意，因为在提高插入记录的速度时，导致查询操作的速度减慢。这就需要找到一个平衡点，比如在设计索引系统时，考虑表的主要功能（数据修复及编辑）不失为一种明智的选择。#2: 优化查询性能 在分析查询性能时，考虑EXPLAIN关键字同样很管用。EXPLAIN关键字一般放在SELECT查询语句的前面，用于描述MySQL如何执行查询操作、以及MySQL成功返回结果集需要执行的行数。下面的一个简单例子可以说明（列表C）这一过程：列表 Cmysql> EXPLAIN SELECT city.name, city.district FROM city, country WHERE city.countrycode = country.code AND country.code = IND. ---- ------------ -------- ------- ------------- -------- -------- ------ ------ ------------ | id | 0select_type | table | type | possible_keys | key | key_len | ref | rows | Extra | ---- ------------ -------- ------ ------------- ---------

-------- ------ ----- ------------ | 1 | SIMPLE | country | const | PRIMARY | PRIMARY | 3 | const | 1 | Using index || 1 | SIMPLE | city | ALL | NULL | NULL | NULL | NULL | 4079 | Using where | ---- ----------- -------- ------ ------------- -------- ---------- ------ ----- ------------ 2 rows in set (0.00 sec)这里查询是基于两个表连接。EXPLAIN关键字描述了MySQL是如何处理连接这两个表。必须清楚的是，当前设计要求MySQL处理的是country表中的一条记录以及city表中的整个4019条记录。这就意味着，还可使用其他的优化技巧改进其查询方法。例如，给city表添加如下索引（列表D）：列表 Dmysql> CREATE INDEX idx_ccode ON city(countrycode).Query OK, 4079 rows affected (0.15 sec)Records: 4079 Duplicates: 0 Warnings: 0现在，当我们重新使用EXPLAIN关键字进行查询时，我们可以看到一个显着的改进（列表E）：列表 Emysql> EXPLAIN SELECT city.name, city.district FROM city, country WHERE city.countrycode = country.code AND country.code = IND. ---- ------------ -------- ------ ------------- ---------- ---------- ------ ----- ------------ | id | 0select_type | table | type | possible_keys | key | key_len | ref | rows | Extra | ---- ------------ -------- ------ ------------- ---------- -------- ------ ------ ------------ | 1 | SIMPLE | country | const | PRIMARY | PRIMARY | 3 | const | 1 | Using index || 1 | SIMPLE | city | ref | idx_ccode | idx_ccode | 3 | const | 333 | Using where | ---- ------------ ---------- ------ ------------- ---------- -------- ------ ------ ------------ 2 rows in set (0.01 sec) 在这个例子中，MySQL现在只需要扫描city表中的333条记录就可产生一个结果集，其扫

描记录数几乎减少了90%！自然，数据库资源的查询速度更快，效率更高。#3: 调整内部变量 MySQL是如此的开放，所以可轻松地进一步调整其缺省设置以获得更优的性能及稳定性。需要优化的一些关键变量如下：改变索引缓冲区长度(key_buffer) 一般，该变量控制缓冲区的长度在处理索引表（读/写操作）时使用。MySQL使用手册指出该变量可以不断增加以确保索引表的最佳性能，并推荐使用与系统内存25%的大小作为该变量的值。这是MySQL十分重要的配置变量之一，如果你对优化和提高系统性能有兴趣，可以从改变key_buffer_size变量的值开始。改变表长(read_buffer_size) 当一个查询不断地扫描某一个表，MySQL会为它分配一段内存缓冲区。read_buffer_size变量控制这一缓冲区的大小。如果你认为连续扫描进行得太慢，可以通过增加该变量值以及内存缓冲区大小提高其性能。设定打开表的数目的最大值(table_cache) 该变量控制MySQL在任何时候打开表的最大数目，由此能控制服务器响应输入请求的能力。它跟max_connections变量密切相关，增加table_cache值可使MySQL打开更多的表，就如增加max_connections值可增加连接数一样。当收到大量不同数据库及表的请求时，可以考虑改变这一值的大小。对缓长查询设定一个时间限制(long_query_time) MySQL带有“ 慢查询日志 ”，它会自动地记录所有的在一个特定的时间范围内尚未结束的查询。这个日志对于跟踪那些低效率或者行为不端的查询以及寻找优化对象都非常有用。long_query_time变量控制这一最大时间限定，以秒为单位。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com