

[博客园](#) [首页](#) [新随笔](#) [联系](#) [订阅](#) [管理](#)

随笔 - 112 文章 - 0 评论 - 168 阅读 - 189万

MySQL IO线程及相关参数调优

一、关于一个SQL的简单的工作过程

1、工作前提描述

1、启动MySQL，在内存中分配一个大空间innodb_buffer_pool(还有log_buffer)

2、多用户线程连接MySQL，从内存分配用户工作空间(其中排序空间)

3、磁盘上有数据库文件、ib_logfile、tmp目录、undo

2、SQL的简易流程

1、DQL操作

1、首先进行内存读

2、如果buffer pool中没有所需数据，就进行物理读

3、物理读数据读入buffer pool，再返回给用户工作空间

2、DML操作(例update)

1、内存读，然后进行物理读，读取所需修改的数据行

2、从磁盘调入undo页到buffer pool中

3、修改前的数据存入undo页里，产生redo

4、修改数据行(buffer pool中数据页成脏页)，产生redo

5、生成的redo先是存于用户工作空间，择机拷入log_buffer中

6、log线程不断的将log_buffer中的记录写入redo logfile中

7、修改完所有数据行，提交事务，刻意再触发一下log线程

8、待log_buffer中的相关信息都写完，响应事务提交成功

至此，日志写入磁盘，内存脏块还在buffer pool中(后台周期写入磁盘，释放buffer pool空间)。

公告

一个人最好的生活状态：该看书时看书，该玩时尽情玩；看见优秀的人欣赏，看见落魄的人也不轻视；有自己的小生活和小情趣，不用去想改变世界，努力去活出自己；没有人爱时专注自己，有人爱时有能力拥抱彼此。

昵称：

GeaoZhang

园龄：5年11个月

粉丝：360

关注：30

[+加关注](#)

最新随笔

[1.MySQL用户也可以是个角色](#)

[2.For Update](#)

[加锁分析](#)

[3.MySQL索引失效之](#)

[4.理](#)

[5.布](#)

[述及](#)

[6.P](#)

[Dat](#)

[7.N](#)

[离级别](#)

二、影响SQL执行性能的因素，及具体看方式

1、大量物理读

```
mysql> show global status like 'i%read%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Innodb_buffer_pool_reads | 64 |
| Innodb_data_read | 48 |
| Innodb_data_reads | 29 |
| Innodb_pages_read | 29 |
| Innodb_rows_read | 10 |
+-----+-----+
```

- 1、Innodb_buffer_pool_reads：物理读次数
- 2、Innodb_data_read：物理读数据字节量
- 3、Innodb_data_reads：物理读IO请求次数
- 4、Innodb_pages_read：物理读数据页数
- 5、Innodb_rows_read：物理读数据行数

2、Log写性能

```
mysql> show engine innodb status \G
---
LOG
---
Log sequence number 144064129 //已经生成的日志
Log flushed up to 144064129 //已经写入的日志
Pages flushed up to 144064129 //已经写入的脏页
Last checkpoint at 144064120 //检查点
0 pending log flushes, 0 pending chkp writes
92 log i/o's done, 0.00 log i/o's/second
```

关于redo log的写入：

- 1、Innodb_os_log_written：日志刷盘的字节数，如果在commit不怎么变化的情况下，这个值出现暴增，说明系统出现大事务了（处理：kill线程，必要情况kill掉mysql进程）；
- 2、Innodb_log_writes：日志写的次数。

3、磁盘排序

```
mysql> show status like 'Sort_merge_passes';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Sort_merge_passes | 0 |
+-----+-----+
```

8.MySQL-[SIGNAL/RESIGNAL/GET DIAGNOSTICS]的使用

9.MySQL的SQL预处理(Prepared)

10.利用PowerShell监控Win-Server性能

积分与排名

积分 - 255007

排名 - 3651

随笔分类 (115)

Algorithm(2)

Computer(10)

Linux(23)

MySQL(67)

OneMore(3)

PowerShell(1)

Python(8)

SQL Server(1)

随笔档案 (112)

2020

2020

2020

2020

2020

2020

2018年10月(1)

2018年3月(2)

```
+-----+-----+
1 row in set (0.00 sec)
```

用户所需数据，如果没有内存buffer pool中，就发生物理读；

如果需要过滤掉很多数据，就会影响物理读和内存读，因为返回很多的数据（物理读），在内存中需要过滤掉很多数据（内存读）；

如果涉及到group/order by，会在用户工作空间完成排序等，如果结果集过大，用户空间过小，进行磁盘排序，Sort_merge_passes>0，这就很影响数据库性能了。

三、MySQL线程及其工作

MySQL的工作机制是单进程多线程：IO线程=一个log线程+四个read线程+四个write线程

```
mysql> show engine innodb status \G
-----
FILE I/O
-----

I/O thread 0 state: waiting for i/o request
I/O thread 1 state: waiting for i/o request
I/O thread 2 state: waiting for i/o request
I/O thread 3 state: waiting for i/o request
I/O thread 4 state: waiting for i/o request
I/O thread 5 state: waiting for i/o request
I/O thread 6 state: waiting for i/o request
I/O thread 7 state: waiting for i/o request
I/O thread 8 state: waiting for i/o request
I/O thread 9 state: waiting for i/o request
```

1、读操作：innodb_read_io_threads

- 1、发起者：用户线程发起读请求
- 2、完成者：读线程执行请求队列中的读请求操作
- 3、如何调整读线程的数量

```
mysql> show variables like 'innodb_read_io_t
+-----+-----+
| Variable_name          | Value |
```

2018年2月(1)
2017年8月(7)
2017年7月(17)
2017年6月(12)
2017年5月(29)
2017年4月(29)
2017年3月(5)

阅读排行榜

- 1. MySQL存储...
- 2. MySQL触发...
- 3. MySQL常用...
- 4. SELECT中...
- 5. MySQL最常...

推荐排行榜

- 1. MySQL存储过程的创建及调用(55)
- 2. MySQL触发器trigger的使用(47)
- 3. 深入解析MySQL视图VIEW(32)
- 4. MySQL最常用分组聚合函数(27)
- 5. i统(

最新

```
+-----+
| innodb_read_io_threads | 4 |
+-----+
1 row in set (0.01 sec)
```

默认是开启4个读线程，静态参数，修改至配置文件中

4、如何确定是否需要增加读线程的数量

查看线程的状态：I/O thread 2 state: waiting for i/o request (read thread)

2、写操作：innodb_write_io_threads

1、发起者：page_cleaner线程发起

2、完成者：写线程执行请求队列中的写请求操作

3、如何调整写线程的数量

```
mysql> show variables like 'innodb_write_io_
+-----+
| Variable_name          | Value |
+-----+
| innodb_write_io_threads | 4 |
+-----+
1 row in set (0.01 sec)
```

默认是开启4个写线程，静态参数，修改至配置文件中

4、如何确定是否需要增加写线程的数量

查看线程的状态：I/O thread 6 state: waiting for i/o request (write thread)

关于innodb_purge_threads: page cleaner 线程

作用:

1、负责对 undo 数据页的清空

2、数据页中 delete 标志行的清除

3、清理 innodb buffer pool，负责把内存中的脏页发起写请求，write 线程负载把脏页刷新到磁盘上。

1. Re:理解代价函数

太强了，听你这么一说，我终于把统计学上的回归和机器学习联系到一起了

--todaroroad

2. Re:Python内存管理机制

窗口对象 为什么叫窗口对象而不是容器对象的名字呢？有出处么？

--Dearning

3. Re:Python内存管理机制

另外的别人被创建 建议改为 被引用

--Dearning

4. Re:MySQL的SQL预处理(Prepared)

学习了，感谢大佬

--星祖666

5. Re:For Update 加锁分析

写的好

-

3、日志线程

3.1、只有一个日志线程

1、是否繁忙

I/O thread 1 state: waiting for i/o request

(log thread): 闲

2、日志写性能

```
mysql> show global status like 'Innodb_log_w
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| Innodb_log_waits   | 0     |
+-----+-----+
1 row in set (0.00 sec)
```

如果log buffer太小，就很容易满，导致无法写入，产生日志等待。

3、日志写压力

1、每秒吞吐量

```
mysql> show global status like 'Innodb_os_lo
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| Innodb_os_log_written | 57856 |
+-----+-----+
1 row in set (0.01 sec)
```

2、每秒写入次数

```
mysql> show global status like 'Innodb_log_w
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| Innodb_log_writes   | 59    |
+-----+-----+
1 row in set (0.01 sec)
```

3.2、对于日志监控来说，三个经典参数

2

- 1、Innodb_log_waits #redo写入的等待次数
- 2、Innodb_log_writes #redo写入的次数

3、Innodb_os_log_written #写入redo logfile

中的字节量

3.3、日志写入异常判断

```
mysql> show engine innodb status \G
Pending flushes (fsync) log: 0;

mysql> show global status like 'Innodb_os_lo
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| Innodb_os_log_pending_fsyncs | 0     |
+-----+-----+
1 row in set (0.00 sec)
```

1、**fsync**：绕过文件系统缓存，直接将内存中的数据写入存储中，实现数据真正写入可靠的介质磁盘里。(对于redo log来说，通过fsync方式写入磁盘才是可靠的保证，因为写入文件系统缓存的提交成功响应并不是真正的将redo写入磁盘的logfile中)(sync:同步)

2、**pending**：挂起(写不动)，redo写入存储cache过程中，某种原因io繁忙，cache被占满，超时响应，就会被挂起；

3、>0就说明系统IO出现问题，=0说明is OK。

四、log buffer调整依据

log buffer：日志缓存，一般都很小，调整100M足够使用

```
mysql> show variables like "%log_buffer%";
+-----+-----+
| Variable_name          | Value |
+-----+-----+
| innodb_log_buffer_size | 16777216 |
+-----+-----+
```

5.7默认是16M，5.6默认是8M

1、文件中(log file)：每次写的时候全局都写，不会挑着捡着写

2

1、日志写线程每一秒redo日志缓冲刷新到重做日志文件

2、每个事务提交时会将重做日志缓冲刷新到重做日志文件

3、每当重做日志缓冲池剩余空间小于1/2时，重做日志缓冲刷新到重做日志文件。

2、在企业中往往设置 50-100M，最多设置为 2、300M，调整依据：

1、内存空间足够大

2、日志产生量大，系统io阻塞了，系统的io占用的是一个带宽，log_writes线程被阻塞，log buffer满了，数据库会hang住。

3、Innodb_log_waits（状态值）

The number of times that the log buffer was too small and a wait was required for it to be flushed before continuing.

@author: <http://www.cnblogs.com/geaozhang/>

分类: [MySQL](#)

好文要顶

关注我

收藏该文



GeaoZhang

粉丝 - 360 关注 - 30

+加关注

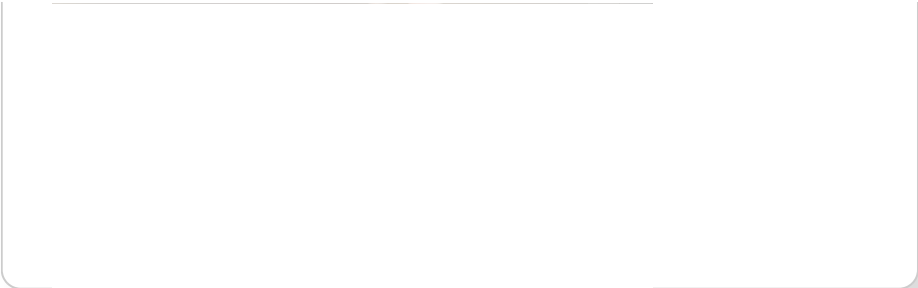
« 上一篇: [MySQL存储写入性能严重抖动分析](#)

» 下一篇: [MySQL后台线程的清理工作](#)

posted @ 2017-07-20 20:08 [GeaoZhang](#) 阅读(8160) 评论(1) [编辑](#) [收藏](#) [举报](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页



Copyright © 2023 GeaoZhang
Powered by .NET 7.0 on Kubernetes