

默认配置下，为什么 MySQL 8.0 比 MySQL 5.7 慢？

原创 Yangsh888 老杨漫谈 2025年02月14日 22:18 广东



这两天在整理 MySQL 的资料，然后分别在 MySQL 5.7 和 MySQL 8.0 上做测试，看看这些资料内容在两个版本上是否有差异。

结果在涉及到一个插入数据的场景中，发现两个版本的性能相差很大，都是在一个简单表上通过存储过程来插入100万条记录。两者的性能相差3倍多。

于是就想看看是什么原因导致两者这么大的差异，以及应该做哪些调整才能有性能提升。

首先说明，两个数据库安装在相同型号，相同配置的两台机器上，都采取默认配置，所配置的内存也一致。

缘起

我就是在下面这张表以及存储过程中发现了两者的性能差异。

```
1 CREATE TABLE `t` (  
2   `id` int(11) NOT NULL,  
3   `a` int(11) DEFAULT NULL,  
4   `b` int(11) DEFAULT NULL,  
5   PRIMARY KEY (`id`),  
6   KEY `a` (`a`),  
7   KEY `b` (`b`)  
8 ) ENGINE=InnoDB;  
9 delimiter ;;  
10 create procedure idata()  
11 begin  
12   declare i int;  
13   set i=1;  
14   while(i<=100000)do  
15     insert into t values(i, i, i);  
16     set i=i+1;  
17   end while;  
18 end;;  
19 delimiter ;
```

下面是执行的结果（含有 mysql-5.7 提示符的表示是 5.7 版本）

```
1 mysql-5.7>call idata();  
2 Query OK, 1 row affected (6.07 sec)  
3  
4 mysql-8.0>call idata();  
5 Query OK, 1 row affected (20.06 sec)
```

可以看出，这个性能相差还是很大的。我想应该是 MySQL 8.0 某些默认参数可能设置得不太好，所以导致比较慢。于是我把两者的默认配置中，我觉得的和本次性能相关的差异列出来了，当然我这里是列出两者都有的参数，至于那些只存在 MySQL 8.0 的参数就没有列出了：

配置参数	mysql 5.7 值	mysql 8.0 值
back_log	80	151
event_scheduler	OFF	ON
innodb_autoinc_lock_mode	1	2
innodb_flush_method	-	fsync
innodb_undo_log_truncate	OFF	ON
innodb_undo_tablespaces	0	2
innodb_flush_neighbors	1	0
innodb_max_dirty_pages_pct	75	90
innodb_max_dirty_pages_pct_lwm	0	10
local_infile	ON	OFF
log_error_verbosity	3	2
master_info_repository	FILE	TABLE
max_allowd_packet	4194304	67108864
max_error_count	64	1024
max_length_for_sort_data	1024	4096
open_files_limit	5000	10000
optimizer_trace_max_mem_size	16384	1048576
performance_schema_max_cond_classes	80	150
performance_schema_max_memory_classes	320	450
performance_schema_max_mutex_classes	210	350
performance_schema_max_rwlock_classes	50	60
performance_schema_max_stage_classes	150	175
performance_schema_max_statement_classes	193	219

performance_schema_max_thread_classes	50	100
pseudo_thread_id	19	18
slave_parallel_type	DATABASE	LOGICAL_CLOCK
slave_parallel_workers	0	4
slave_pending_jobs_size_max	16777216	134217728
slave_preserve_commit_order	OFF	ON
table_definition_cache	1400	2000
table_open_cache	2000	4000
thread_stack	262144	1048576
transaction_write_set_extraction	OFF	XXHASH64

排查

既然上面这些参数不同，那挨个把参数的值改成一样，看看性能有什么变化，于是我把上面涉及到 InnoDB 的参数逐个进行调整，得到了下面的这个性能表：

参数	值	5.7 性能(秒)	8.0 性能(秒)
innodb_autoinc_lock_mode	1	6.11	20.32
innodb_undo_log_truncate	OFF	6.10	20.18
innodb_max_dirty_pages_pct	75	6.14	19.89
innodb_max_dirty_pages_pct_lwm	0	6.13	19.87
back_log	80	6.07	20.23

一顿操作下来，发现参数对性能差异没有太多影响，我突然，难道是因为虽然两台机器的配置相同，但可能磁盘的性能不用相同了？于是对两台机器的磁盘进行了一轮性能测试，使用的工具为 fio ，测试的参数如下：

```
1 $ /usr/bin/fio --randrepeat=1 --ioengine=libaio \  
2   --direct=1 --gtod_reduce=1 --name=test \  
3   --filename=test --bs=4k --iodepth=64 \  
4   --size=4G --readwrite=randrw --rwmixread=75
```

10 分钟后，得到的结果是两台机器的硬盘性能基本无差异。

下面是 MySQL 5.7 那台机器的结果：

```
1 Starting 1 process  
2 test: Laying out IO file (1 file / 4096MiB)  
3 Jobs: 1 (f=1): [m(1)][100.0%][r=10.1MiB/s,w=3440KiB/s][r=2586,w=860 IO  
4 test: (groupid=0, jobs=1): err= 0: pid=21548: Sat Mar 25 22:42:23 2023  
5   read: IOPS=1684, BW=6739KiB/s (6901kB/s)(3070MiB/466501msec)  
6   bw (  KiB/s): min= 3904, max=10776, per=100.00%, avg=6737.87, stdev=  
7   iops       : min=  976, max= 2694, avg=1684.45, stdev=157.83, samp  
8   write: IOPS=563, BW=2252KiB/s (2306kB/s)(1026MiB/466501msec)  
9   bw (  KiB/s): min= 1368, max= 3584, per=100.00%, avg=2251.96, stdev=  
10  iops        : min=  342, max=  896, avg=562.96, stdev=57.50, sample  
11  cpu         : usr=1.07%, sys=3.71%, ctx=999168, majf=0, minf=8  
12  IO depths   : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=6  
13    submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,  
14    complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.1%,  
15    issued rwts: total=785920,262656,0,0 short=0,0,0,0 dropped=0,0,0,  
16    latency   : target=0, window=0, percentile=100.00%, depth=64  
17 Run status group 0 (all jobs):  
18   READ: bw=6739KiB/s (6901kB/s), 6739KiB/s-6739KiB/s (6901kB/s-6901kB  
19   WRITE: bw=2252KiB/s (2306kB/s), 2252KiB/s-2252KiB/s (2306kB/s-2306kB  
20 Disk stats (read/write):  
21   sda: ios=785514/263262, merge=0/17, ticks=29813462/19803, in_queue=2
```

这个下面是 MySQL 8.0 那台机器的结果：

```
1 Starting 1 process  
2 test: Laying out IO file (1 file / 4096MiB)  
3 Jobs: 1 (f=1): [m(1)][100.0%][r=9488KiB/s,w=3232KiB/s][r=2372,w=808 IO  
4 test: (groupid=0, jobs=1): err= 0: pid=22261: Sat Mar 25 22:42:21 2023  
5   read: IOPS=1692, BW=6768KiB/s (6931kB/s)(3070MiB/464483msec)  
6   bw (  KiB/s): min= 3216, max=10040, per=99.91%, avg=6762.12, stdev=  
7   iops       : min=  804, max= 2510, avg=1690.51, stdev=145.43, samp  
8   write: IOPS=565, BW=2262KiB/s (2316kB/s)(1026MiB/464483msec)  
9   bw (  KiB/s): min= 1024, max= 3456, per=99.96%, avg=2260.11, stdev=  
10  iops        : min=  256, max=  864, avg=565.00, stdev=53.70, sample  
11  cpu         : usr=1.10%, sys=3.74%, ctx=999072, majf=0, minf=8  
12  IO depths   : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=6  
13    submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,  
14    complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.1%,  
15    issued rwts: total=785920,262656,0,0 short=0,0,0,0 dropped=0,0,0,  
16    latency   : target=0, window=0, percentile=100.00%, depth=64  
17 Run status group 0 (all jobs):  
18   READ: bw=6768KiB/s (6931kB/s), 6768KiB/s-6768KiB/s (6931kB/s-6931kB  
19   WRITE: bw=2262KiB/s (2316kB/s), 2262KiB/s-2262KiB/s (2316kB/s-2316kB  
20 Disk stats (read/write):  
21   sda: ios=785411/262588, merge=0/8, ticks=29677795/20550, in_queue=29
```


可以看到两台硬盘的读的 IOPS 分别是 1684.45 和 1690.5，差异不到 3%。

而写的 IOPS 分别为 562.96 和 565.00，差异率在千分之4以内。

曙光

于是开始疯狂的找资料，然后找到一个参数：

```
1 transaction_write_set_extraction
```

该参数用来指定用于对事务中提取的写入进行哈希的算法，在 MySQL 5.7 默认不启用，而 8.0 则默认配置为 XXHASH64，尝试都改成 OFF，得到如下的结果：

```
1 mysql-5.7>call idata();
2 Query OK, 1 row affected (6.15 sec)
3
4 mysql-8.0>call idata();
5 Query OK, 1 row affected (14.75 sec)
```

这两者的差异一下小了不少，但仍然有2倍的差距，不应该啊。

疏忽

继续折腾，显著性能差异还是在那里，我想不应该啊，如果真的性能差异这么大，社区还不闹翻天了。我想不会是存储的文件格式有些区别，于是去看数据目录，看有什么发现没有，进入 MySQL 8.0 时，咦，怎么看到有 binlog 日志文件？不是默认关闭了吗？

再看 MySQL 5.7，没有 binlog 日志，好家伙，搞得半天还是自己的原因啊。原来 MySQL 5.7 里如果不希望写 binlog，那设置 log_bin=0 或者 log_bin=off 都可以，而 MySQL 8.0 ,则需要配置为 skip-log-bin。

赶紧设置后，再继续测试：

```
1 mysql-5.7>call idata();
2 Query OK, 1 row affected (6.11 sec)
3
4 mysql-8.0>call idata();
5 Query OK, 1 row affected (7.03 sec)
```

这下性能差异在 15% 以内了。考虑到因为 MySQL 8.0 多了很多新特性，估计在开箱即用的情况下，MySQL 8.0 性能打不过 MySQL 5.7。

那就尝试优化看看。

优化

继续调整与 MySQL 8.0 有关的参数，看看在这个纯插入的场景下，MySQL 8.0 是否能和 MySQL 5.7 持平甚至超过，但经过多次参数调整后，目前依然是 MySQL 5.7 获胜。也许在综合负载上，MySQL 8.0 可能胜于 MySQL 5.7，比如我看到有人做的测试中，当并发的线程超过 1024 后，MySQL 8.0 的性能就相比 5.7 有了显著的提升。

后续有时间可以使用 sysbench 之类的工具搞一个综合负载比较。

参考

以下是本文的部分参考资料：

- <https://stackoverflow.com/questions/67685738/mysql-performance-5-7-vs-8-0>
- <https://severalnines.com/blog/mysql-performance-benchmarking-mysql-57-vs-mysql-80/>
- <https://dba.stackexchange.com/questions/216352/inserts-in-mysql-8-are-slower-than-inserts-in-mysql-5-7>
- <https://bugs.mysql.com/bug.php?id=93734>
- http://dimitrik.free.fr/Presentations/MySQL_Tuning-Feb2019-dim.pdf
- <https://www.devart.com/dbforge/mysql/studio/mysql-performance-tips.html>



老杨漫谈

究天人之际，通古今之变，成一家之言。

121篇原创内容



公众号

运维日记 15

运维日记 · 目录 ≡

◀ 上一篇 · 通过配置 Nginx 实现仅获取可信来源的真实 IP 请求头

个人观点，仅供参考