

比较全面的MySQL优化参考（上篇）

本文整理了一些MySQL的通用优化方法，做个简单的总结分享，旨在帮助那些没有专职MySQL DBA的企业做好基本的优化工作，至于具体的SQL优化，大部分通过加适当的索引即可达到效果，更复杂的就需要具体分析了，可以参考本站的一些优化案例或者联系我，下方有我的联系方式。这是上篇。

1、硬件层相关优化

1.1、CPU相关

在服务器的BIOS设置中，可调整下面的几个配置，目的是发挥CPU最大性能，或者避免经典的NUMA问题：

- 1、选择Performance Per Watt Optimized(DAPC)模式，发挥CPU最大性能，跑DB这种通常需要高运算量的服务就不要考虑节电了；
- 2、关闭C1E和C States等选项，目的也是为了提升CPU效率；
- 3、Memory Frequency（内存频率）选择Maximum Performance（最佳性能）；
- 4、内存设置菜单中，启用Node Interleaving，避免NUMA问题；

1.2、磁盘I/O相关

下面几个是按照IOPS性能提升的幅度排序，对于磁盘I/O可优化的一些措施：

- 1、使用SSD或者PCIe SSD设备，至少获得数百倍甚至万倍的IOPS提升；确保意外时不至于丢失数据
- 2、购置阵列卡同时配备CACHE及BBU模块，可明显提升IOPS（主要是指机械盘，SSD或PCIe SSD除外。同时需要定期检查CACHE及BBU模块的健康状况，确保意外时不至于丢失数据）
- 3、有阵列卡时，设置阵列写策略为WB，甚至FORCE WB（若有双电保护，或对数据安全性要求不是特别高的话），严禁使用WT策略。并且闭阵列预读策略，基本上是鸡肋；闭阵列预读策略，基本上是鸡肋，用处不大
- 4、尽可能选用RAID-10，而非RAID-5；
- 5、使用机械盘的话，尽可能选择高转速的，例如选用15KRPM，而不是7.2KRPM的盘，不差几个钱的；

2、系统层相关优化

2.1、文件系统层优化

在文件系统层，下面几个措施可明显提升IOPS性能：

- 1、使用deadline/noop这两种I/O调度器，千万别用cfq（它不适合跑DB类服务）；
- 2、使用xfs文件系统，千万别用ext3；ext4勉强可用，但业务量很大的话，则一定要用xfs；
- 3、文件系统mount参数中增加：noatime, nodiratime, nobarrier几个选项（nobarrier是xfs文件系统特有的）；

2.2、其他内核参数优化

针对关键内核参数设定合适的值，目的是为了减少swap的倾向，并且让内存和磁盘I/O不会出现大幅波动，导致瞬间波峰负载：

- 1、将vm.swappiness设置为5-10左右即可，甚至设置为0（RHEL 7以上则慎重设置为0，除非你允许OOM kill发生），以降低使用SWAP的机会；
- 2、将vm.dirty_background_ratio设置为5-10，将vm.dirty_ratio设置为它的两倍以上，以确保能持续将脏数据刷新到磁盘，避免瞬间I/O写，产生严重等待（和MySQL中的innodb_max_dirty_pages_pct类似）
- 3、将net.ipv4.tcp_tw_recycle、net.ipv4.tcp_tw_reuse都设置为1，减少TIME_WAIT，提高TCP效率；

4、至于网传的read_ahead_kb、nr_requests这两个参数，我经过测试后，发现对读写混合为主的OLTP环境影响并不大（应该是对读敏感的场景更有效果），不过没准是我测试方法有问题，可自行斟酌是否调整

后记：本文根据个人多年经验总结，个别建议可能有不完善之处，欢迎留言或者加我 微信公众号：[MySQL中文网](#)、QQ：[4700963](#) 相互探讨交流。

附录：延伸阅读

- 1、[常用PC服务器阵列卡](#)、[硬盘健康监控](#)
- 2、[PC服务器阵列卡管理简易手册](#)
- 3、[实测Raid5 VS Raid1+0下的innodb性能](#)
- 4、[SAS vs SSD各种模式下MySQL TPCC OLTP对比测试结果](#)
- 5、[MySQL出了门，Percona在左，MariaDB在右](#)
- 6、[Percona Thread Pool性能基准测试](#)
- 7、[\[MySQL优化案例\]系列 — 分页优化](#)
- 8、[\[MySQL FAQ\]系列 — 为什么InnoDB表要建议用自增列做主键](#)
- 9、[\[MySQL FAQ\]系列 — 为什么要关闭query cache，如何关闭](#)

更多相关优化案例，可在 [百度](#) 或 [谷歌](#) 中根据关键字搜索本站。

比较全面的MySQL优化参考（下篇）

本文整理了一些MySQL的通用优化方法，做个简单的总结分享，旨在帮助那些没有专职MySQL DBA的企业做好基本的优化工作，至于具体的SQL优化，大部分通过加适当的索引即可达到效果，更复杂的就需要具体分析了，可以参考本站的一些优化案例或者联系我，下方有我的联系方式。这是下篇。

3、MySQL层相关优化

3.1、关于版本选择

官方版本我们称为ORACLE MySQL，这个没什么好说的，相信绝大多数人会选择它。

我个人强烈建议选择Percona分支版本，它是一个相对比较成熟的、优秀的MySQL分支版本，在性能提升、可靠性、管理型方面做了不少改善。它和官方ORACLE MySQL版本基本完全兼容，并且性能大约有20%以上的提升，因此我优先推荐它，我自己也从2008年一直以它为主。

另一个重要的分支版本是MariaDB，说MariaDB是分支版本其实已经不太合适了，因为它的目标是取代ORACLE MySQL。它主要在原来的MySQL Server层做了大量的源码级改进，也是一个非常可靠的、优秀的分支版本。但也由此产生了以GTID为代表的和官方版本无法兼容的新特性（MySQL 5.7开始，也支持GTID模式在线动态开启或关闭了），也考虑到绝大多数人还是会跟着官方版本走，因此没优先推荐MariaDB。

3.2、关于最重要的参数选项调整建议

建议调整下面几个关键参数以获得较好的性能（可使用本站提供的my.cnf生成器生成配置文件模板）：

- 此外，还有extra_port功能，非常实用，关键时刻能救命的。还有另外一个重要特色是 QUERY_RESPONSE_TIME 功能，也能使我们对整体的SQL响应时间分布有直观感受；
- 1、选择Percona或MariaDB版本的话，强烈建议启用thread pool特性，可使得在高并发的情况下，性能不会发生大幅下降。此外，还有extra_port功能，非常实用；
 - 2、设置default-storage-engine=InnoDB，也就是默认采用InnoDB引擎，强烈建议不要再使用MyISAM引擎了，InnoDB引擎绝对可以满足99%以上的业务场景；
 - 3、调整innodb_buffer_pool_size大小，如果是单实例且绝大多数是InnoDB引擎表的话，可考虑设置为物理内存的50% ~ 70%左右；
 - 4、根据实际需要设置innodb_flush_log_at_trx_commit、sync_binlog的值。如果要求数据不能丢失，那么两个都设为1。如果允许丢失一点数据，则可分别设为2和10。而如果完全不用care数据是否丢失的话（例如在slave上，反正大不了重做一次），则可都设为0。这三种设置值导致数据库的性能受到影响程度分别是：高、中、低，也就是第一个会另数据库最慢，最后一个则相反；
 - 5、设置innodb_file_per_table = 1，使用独立表空间，我实在是想不出来用共享表空间有什么好处了；
 - 6、设置innodb_data_file_path = ibdata1:1G:autoextend，千万不要用默认的10M，否则在有高并发事务时，会受到不小的影响；
 - 7、设置innodb_log_file_size=256M，设置innodb_log_files_in_group=2，基本可满足90%以上的场景；
 - 8、设置long_query_time = 1，而在5.5版本以上，已经可以设置为小于1了，建议设置为0.05（50毫秒），记录那些执行较慢的SQL，用于后续的分析排查；
 - 9、根据业务实际需要，适当调整max_connection（最大连接数）、max_connection_error（最大错误数，建议设置为10万以上，而open_files_limit、innodb_buffer_pool_size、innodb_open_files、table_open_cache、table_definition_cache这几个参数则可设为约10倍于max_connection的大小；
 - 10、常见的误区是把tmp_table_size和max_heap_table_size设置的比较大，曾经见过设置为1G的，这两个选项是每个连接会话都会分配的，因此不要设置过大，否则容易导致OOM发生；其他的一些连接会话级选项例如：sort_buffer_size、join_buffer_size、read_buffer_size、read_rnd_buffer_size等，也需要注意不能设置过大；
 - 11、由于已经建议不再使用MyISAM引擎了，因此可以把key_buffer_size设置为32M左右，并且强烈建议关闭query_cache功能；

3.3、关于Schema设计规范及SQL使用建议

下面列举了几个常见有助于提升MySQL效率的Schema设计规范及SQL使用建议：

- 1、所有的InnoDB表都设计一个无业务用途的自增列做主键，对于绝大多数场景都是如此，真正纯只读用InnoDB表的并不多，真如此的话还不如用TokuDB来得划算；
- 2、字段长度满足需求前提下，尽可能选择长度小的。此外，字段属性尽量都加上NOT NULL约束，可一定程度提高性能；
- 3、尽可能不使用TEXT/BLOB类型，确实需要的话，建议拆分到子表中，不要和主表放在一起，避免SELECT * 的时候读性能太差。
- 4、读取数据时，只选取所需要的列，不要每次都SELECT *，避免产生严重的随机读问题，尤其是读到一些TEXT/BLOB列；

- 5、对一个VARCHAR(N)列创建索引时，通常取其50%（甚至更小）左右长度创建前缀索引就足以满足80%以上的查询需求了，没必要创建整列的全长度索引；
- 6、通常情况下，子查询的性能比较差，建议改造成JOIN写法；
- 7、多表联接查询时，关联字段类型尽量一致，并且都要有索引；
- 8、多表连接查询时，把结果集小的表（注意，这里是指过滤后的结果集，不一定是全表数据量小的）作为驱动表；
- 9、多表联接并且有排序时，排序字段必须是驱动表里的，否则排序列无法用到索引；
- 10、多用复合索引，少用多个独立索引，尤其是一些基数（Cardinality）太小（比如说，该列的唯一值总数少于255）的列就不要创建独立索引了；
- 11、类似分页功能的SQL，建议先用主键关联，然后返回结果集，效率会高很多；

3.4、其他建议

关于MySQL的管理维护的其他建议有：

- 不用担心性能问题，这么建议主要是考虑ONLINE DDL的代价较高；
- 1、通常地，单表物理大小不超过10GB，单表行数不超过1亿条，行平均长度不超过8KB，如果机器性能足够，这些数据量MySQL是完全能处理的过来的，不用担心性能问题；
 - 2、不用太担心mysqld进程占用太多内存，只要不发生OOM kill和用到大量的SWAP都还好；
 - 3、在以往，单机上跑多实例的目的是能最大化利用计算资源，如果单实例已经能耗尽大部分计算资源的话，就没必要再跑多实例了；
 - 4、定期使用pt-duplicate-key-checker检查并删除重复的索引。定期使用pt-index-usage工具检查并删除使用频率很低的索引；
 - 5、定期采集slow query log，用pt-query-digest工具进行分析，可结合Anemometer系统进行slow query管理以便分析slow query并进行后续优化工作；
 - 6、可使用pt-kill杀掉超长时间的SQL请求，Percona版本中有个选项 innodb_kill_idle_transaction 也可实现该功能；
 - 7、使用pt-online-schema-change来完成大表的ONLINE DDL需求；
 - 8、定期使用pt-table-checksum、pt-table-sync来检查并修复mysql主从复制的数据差异；

后记：本文根据个人多年经验总结，个别建议可能有不完善之处，欢迎留言或者加我 微信公众号：[MySQL中文网](#)、QQ：[4700963](#) 相互探讨交流。

写在最后：这次的优化参考，大部分情况下我都介绍了适用的场景，如果你的应用场景和本文描述的不太一样，那么建议根据实际情况进行调整，而不是生搬硬套。欢迎质疑拍砖，但拒绝不经过大脑的习惯性抵制。

附录：延伸阅读

- 1、[常用PC服务器阵列卡、硬盘健康监控](#)
- 2、[PC服务器阵列卡管理简易手册](#)
- 3、[实测Raid5 VS Raid1+0下的innodb性能](#)
- 4、[SAS vs SSD各种模式下MySQL TPCC OLTP对比测试结果](#)
- 5、[MySQL出了门，Percona在左，MariaDB在右](#)
- 6、[Percona Thread Pool性能基准测试](#)
- 7、[\[MySQL优化案例\]系列 — 分页优化](#)
- 8、[\[MySQL FAQ\]系列 — 为什么InnoDB表要建议用自增列做主键](#)
- 9、[\[MySQL FAQ\]系列 — 为什么要关闭query cache，如何关闭](#)

更多相关优化案例，可在 [百度](#) 或 [谷歌](#) 中根据关键字搜索本站。

本条目发布于2015/05/29。属于数据库分类，被贴了 [MySQL](#)、[MySQL优化](#) 标签。

LAN

2015/06/024:40 下午

楼主对percona 工具有情有独钟呀。。

[回复](#) ↓

yejr 文章作者

2015/06/0211:45 下午

[确实方便实用](#)

[回复](#) ↓

1

2015/06/1611:41 上午

11、由于已经建议不再使用MyISAM引擎了，因此可以把key_buffer_size设置为32M左右，并且强烈建议关闭query cache功能

[针对innodb也不建议开启吗](#)

[回复](#) ↓

yejr 文章作者

2015/06/178:11 上午

[是的，一律关闭](#)，具体可以搜本站相关文章

[回复](#) ↓