SQL优化不会？推荐4 款工具

爱码士人员 人工智能与大数据技术 2021-12-01 19:18

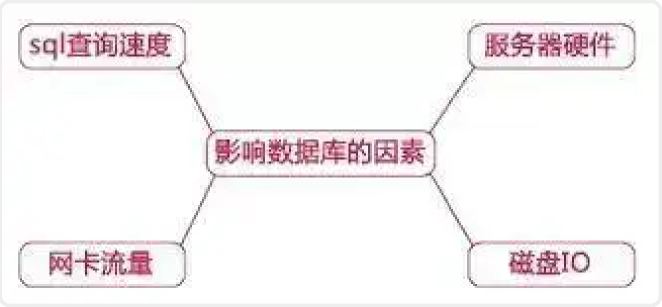
> **来自：简书，作者：爱码士人员**
> **链接：https://www.jianshu.com/p/cb2be017d5a9**

对于正在运行的mysql 性能如何？参数设置的是否合理？账号设置的是否存在安全隐患？

是否了然于胸？

俗话说工欲善其事，必先利其器，定期对你的MYSQL数据库进行一个体检，是保证数据库安全运行的重要手段。

今天和大家分享几个mysql 优化的工具，你可以使用它们对你的mysql进行一个体检，生成awr报告，让你从整体上把握你的数据库的性能情况。



1、mysqltuner.pl

这是mysql一个常用的数据库性能诊断工具，主要检查参数设置的合理性包括日志文件、存储引擎、安全建议及性能分析。针对潜在的问题，给出改进的建议，是mysql优化的好帮手。

在上一版本中，MySQLTuner支持MySQL / MariaDB / Percona Server的约300个指标。

项目地址：https://github.com/major/MySQLTuner-perl

1.1 下载

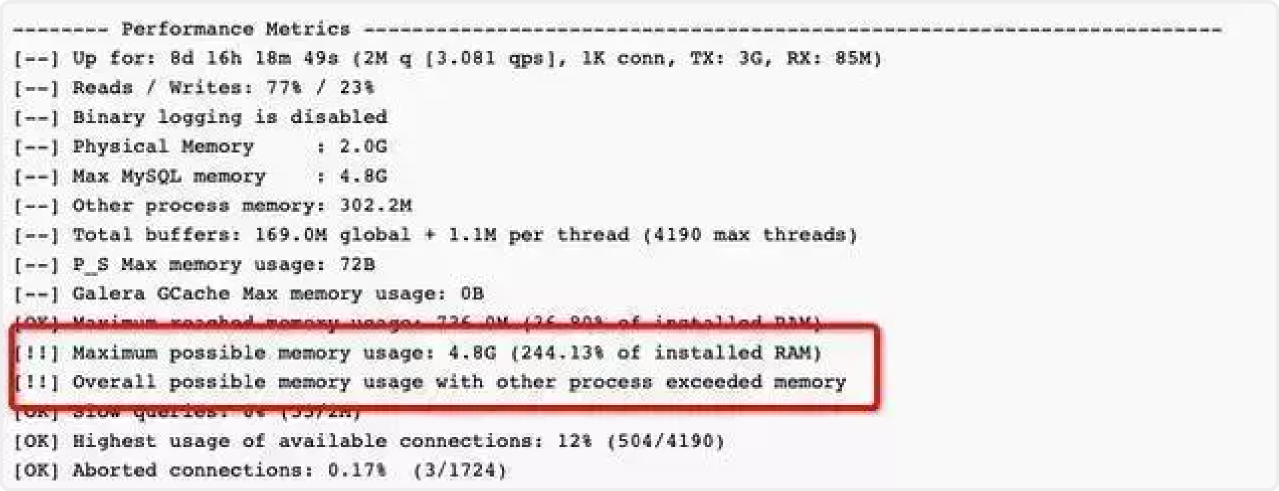
```
[root@localhost ~]#wget https://raw.githubusercontent.com/major/MySQLTuner-perl/master/my
```

1.2 使用

```
[root@localhost ~]# ./mysqltuner.pl --socket /var/lib/mysql/mysql.sock
>> MySQLTuner1.7.4- MajorHayden<major@mhtx.net>
>> Bug reports, feature requests, and downloads at http://mysqltuner.com/
>> Runwith'--help'for additional options and output filtering
[--] Skipped version check forMySQLTuner script
Please enter your MySQL administrative login: root
Please enter your MySQL administrative password: [OK] Currently running supported MySQL v
[OK] Operating on 64-bit architecture
```

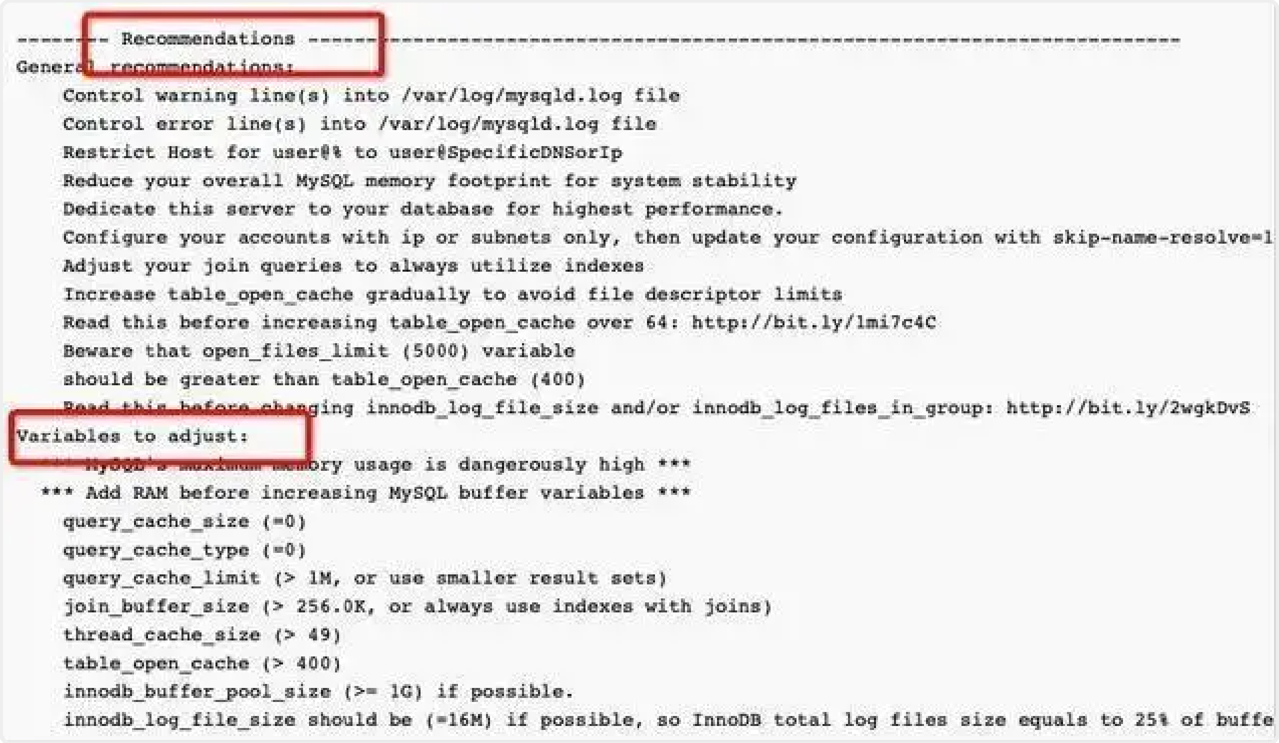
1.3、报告分析

1）重要关注[!!]（中括号有叹号的项）例如[!!] Maximum possible memory usage: 4.8G (244.13% of installed RAM)，表示内存已经严重用超了。



```
-------- Performance Metrics -------------------------------------------------------------------
[--] Up for: 8d 16h 18m 49s (2M q [3.081 qps], 1K conn, TX: 3G, RX: 85M)
[--] Reads / Writes: 77% / 23%
[--] Binary logging is disabled
[--] Physical Memory     : 2.0G
[--] Max MySQL memory    : 4.8G
[--] Other process memory: 302.2M
[--] Total buffers: 169.0M global + 1.1M per thread (4190 max threads)
[--] P_S Max memory usage: 72B
[--] Galera GCache Max memory usage: 0B
[OK] Maximum reached memory usage: 736.0M (36.80% of installed RAM)
[!!] Maximum possible memory usage: 4.8G (244.13% of installed RAM)
[!!] Overall possible memory usage with other process exceeded memory
[OK] Slow queries: 0% (35/2M)
[OK] Highest usage of available connections: 12% (504/4190)
[OK] Aborted connections: 0.17%  (3/1724)
```

2）关注最后给的建议“Recommendations ”。



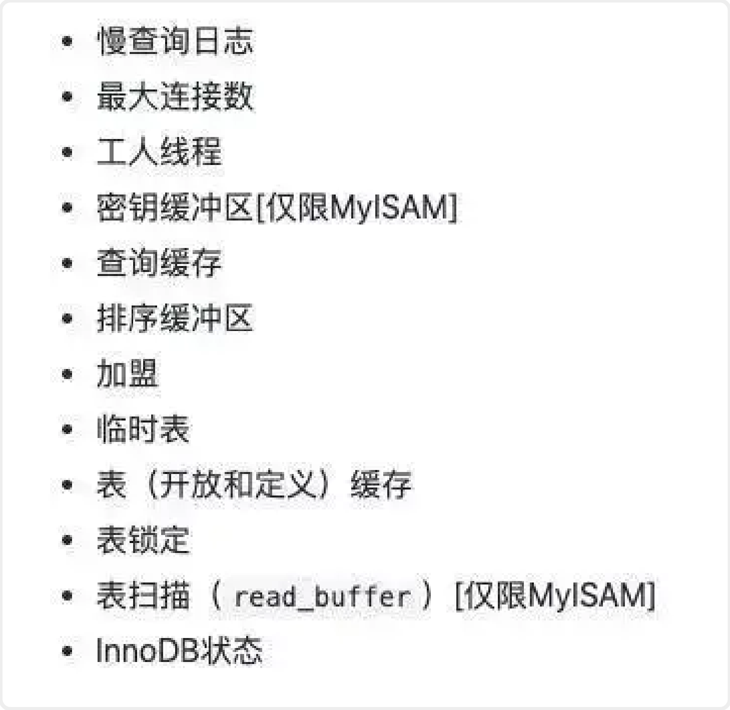
```
-------- Recommendations ---------------------------------------------------------------------
General recommendations:
    Control warning line(s) into /var/log/mysqld.log file
    Control error line(s) into /var/log/mysqld.log file
    Restrict Host for user@% to user@SpecificDNSorIp
    Reduce your overall MySQL memory footprint for system stability
    Dedicate this server to your database for highest performance.
    Configure your accounts with ip or subnets only, then update your configuration with skip-name-resolve=1
    Adjust your join queries to always utilize indexes
    Increase table_open_cache gradually to avoid file descriptor limits
    Read this before increasing table_open_cache over 64: http://bit.ly/1mi7c4C
    Beware that open_files_limit (5000) variable
    should be greater than table_open_cache (400)
    Read this before changing innodb_log_file_size and/or innodb_log_files_in_group: http://bit.ly/2wgkDvS
Variables to adjust:
  *** MySQL's maximum memory usage is dangerously high ***
  *** Add RAM before increasing MySQL buffer variables ***
    query_cache_size (=0)
    query_cache_type (=0)
    query_cache_limit (> 1M, or use smaller result sets)
    join_buffer_size (> 256.0K, or always use indexes with joins)
    thread_cache_size (> 49)
    table_open_cache (> 400)
    innodb_buffer_pool_size (>= 1G) if possible.
    innodb_log_file_size should be (=16M) if possible, so InnoDB total log files size equals to 25% of buffe
```

2、tuning-primer.sh

这是mysql的另一个优化工具，针于mysql的整体进行一个体检，对潜在的问题，给出优化的建议。

项目地址：https://github.com/BMDan/tuning-primer.sh

目前，支持检测和优化建议的内容如下：



- 慢查询日志
- 最大连接数
- 工人线程
- 密钥缓冲区[仅限MyISAM]
- 查询缓存
- 排序缓冲区
- 加盟
- 临时表
- 表（开放和定义）缓存
- 表锁定
- 表扫描（ read_buffer ）[仅限MyISAM]
- InnoDB状态

2.1 下载

```
[root@localhost ~]#wget https://launchpad.net/mysql-tuning-primer/trunk/1.6-r1/+download/
```

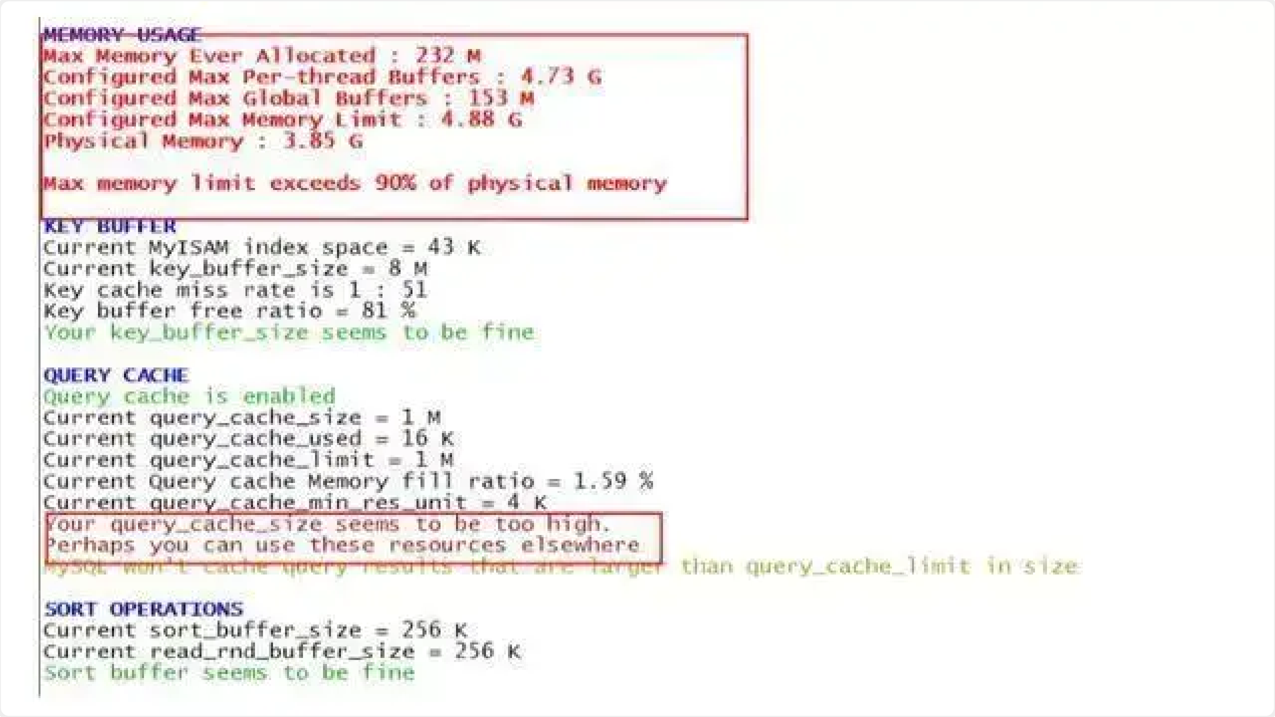
2.2 使用

```
[root@localhost ~]# [root@localhost dba]# ./tuning-primer.sh

-- MYSQL PERFORMANCE TUNING PRIMER --
- By: MatthewMontgomery-
```

2.3 报告分析

重点查看有红色告警的选项，根据建议结合自己系统的实际情况进行修改，例如：



```
MEMORY USAGE
Max Memory Ever Allocated : 232 M
Configured Max Per-thread Buffers : 4.73 G
Configured Max Global Buffers : 153 M
Configured Max Memory Limit : 4.88 G
Physical Memory : 3.85 G

Max memory limit exceeds 90% of physical memory

KEY BUFFER
Current MyISAM index space = 43 K
Current key_buffer_size = 8 M
Key cache miss rate is 1 : 51
Key buffer free ratio = 81 %
Your key_buffer_size seems to be fine

QUERY CACHE
Query cache is enabled
Current query_cache_size = 1 M
Current query_cache_used = 16 K
Current query_cache_limit = 1 M
Current Query cache Memory fill ratio = 1.59 %
Current query_cache_min_res_unit = 4 K
Your query_cache_size seems to be too high.
Perhaps you can use these resources elsewhere
MySQL won't cache query results that are larger than query_cache_limit in size

SORT OPERATIONS
Current sort_buffer_size = 256 K
Current read_rnd_buffer_size = 256 K
Sort buffer seems to be fine
```

3、pt-variable-advisor

pt-variable-advisor 可以分析MySQL变量并就可能出现的问题提出建议。

3.1 安装

https://www.percona.com/downloads/percona-toolkit/LATEST/

```
[root@localhost ~]#wget https://www.percona.com/downloads/percona-toolkit/3.0.13/binary/r
[root@localhost ~]#yum install percona-toolkit-3.0.13-1.el7.x86_64.rpm
```

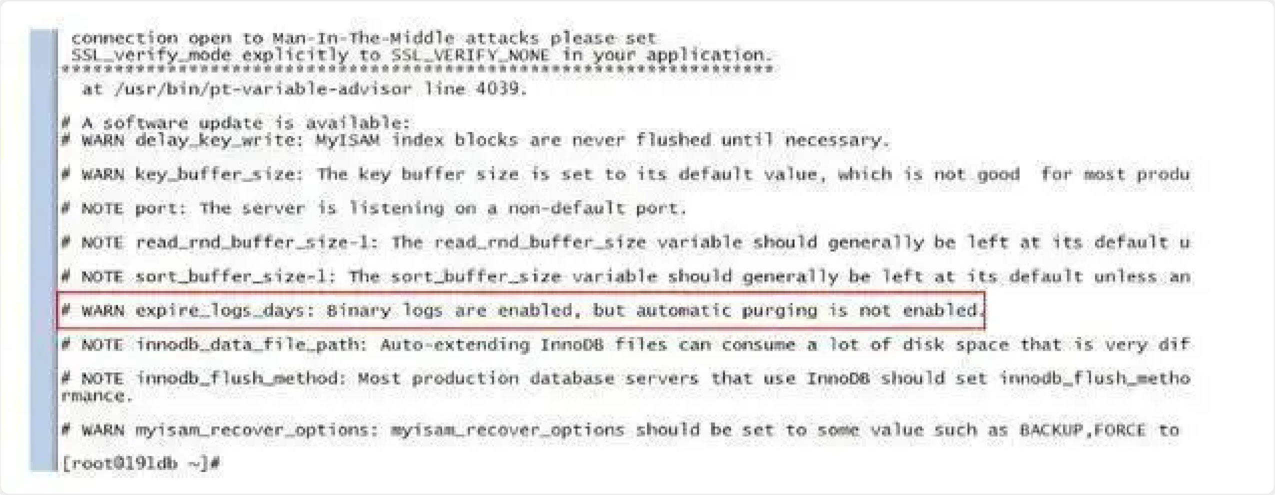
3.2 使用

pt-variable-advisor是pt工具集的一个子工具，主要用来诊断你的参数设置是否合理。

```
[root@localhost ~]# pt-variable-advisor localhost --socket /var/lib/mysql/mysql.sock
```

3.3 报告分析

重点关注有WARN的信息的条目，例如：



```
 connection open to Man-In-The-Middle attacks please set
 SSL_verify_mode explicitly to SSL_VERIFY_NONE in your application.
*******************************************************************
  at /usr/bin/pt-variable-advisor line 4039.

# A software update is available:
# WARN delay_key_write: MyISAM index blocks are never flushed until necessary.

# WARN key_buffer_size: The key buffer size is set to its default value, which is not good  for most produ

# NOTE port: The server is listening on a non-default port.

# NOTE read_rnd_buffer_size-1: The read_rnd_buffer_size variable should generally be left at its default u

# NOTE sort_buffer_size-1: The sort_buffer_size variable should generally be left at its default unless an

# WARN expire_logs_days: Binary logs are enabled, but automatic purging is not enabled.

# NOTE innodb_data_file_path: Auto-extending InnoDB files can consume a lot of disk space that is very dif

# NOTE innodb_flush_method: Most production database servers that use InnoDB should set innodb_flush_metho
rmance.

# WARN myisam_recover_options: myisam_recover_options should be set to some value such as BACKUP,FORCE to

[root@191db ~]#
```

4、pt-qurey-digest

pt-query-digest 主要功能是从日志、进程列表和tcpdump分析MySQL查询。

4.1安装

具体参考3.1节

4.2使用

pt-query-digest主要用来分析mysql的慢日志，与mysqldumpshow工具相比，py-query_digest 工具的分析结果更具体，更完善。

```
[root@localhost ~]# pt-query-digest /var/lib/mysql/slowtest-slow.log
```

4.3 常见用法分析

1）直接分析慢查询文件:

```
pt-query-digest /var/lib/mysql/slowtest-slow.log > slow_report.log
```

2）分析最近12小时内的查询：

```
pt-query-digest --since=12h/var/lib/mysql/slowtest-slow.log > slow_report2.log
```

3）分析指定时间范围内的查询：

```
pt-query-digest /var/lib/mysql/slowtest-slow.log --since '2017-01-07 09:30:00'--until'201
```

4）分析指含有select语句的慢查询

```
pt-query-digest --filter '$event->{fingerprint} =~ m/^select/i'/var/lib/mysql/slowtest-sl
```

```
pt-query-digest --filter '$event->{fingerprint} =~ m/^select/i' /var/lib/mysql/slowtest-sl
```

5）针对某个用户的慢查询

```
pt-query-digest --filter '($event->{user} || "") =~ m/^root/i'/var/lib/mysql/slowtest-slo
```

6）查询所有所有的全表扫描或full join的慢查询

```
pt-query-digest --filter '(($event->{Full_scan} || "") eq "yes") ||(($event->{Full_join}
```

4.4 报告分析

- 第一部分：总体统计结果 Overall：总共有多少条查询 Time range：查询执行的时间范围 unique：唯一查询数量，即对查询条件进行参数化以后，总共有多少个不同的查询 total：总计 min：最小 max：最大 avg：平均 95%：把所有值从小到大排列，位置位于95%的那个数，这个数一般最具有参考价值 median：中位数，把所有值从小到大排列，位置位于中间那个数

- 第二部分：查询分组统计结果 Rank：所有语句的排名，默认按查询时间降序排列，通过--order-by指定 Query ID：语句的ID，（去掉多余空格和文本字符，计算hash值） Response：总的响应时间 time：该查询在本次分析中总的时间占比 calls：执行次数，即本次分析总共有多少条这种类型的查询语句 R/Call：平均每次执行的响应时间 V/M：响应时间Variance-to-mean的比率 Item：查询对象

- 第三部分：每一种查询的详细统计结果 ID：查询的ID号，和上图的Query ID对应 Databases：数据库名 Users：各个用户执行的次数（占比） Query_time distribution ：查询时间分布，长短体现区间占比。Tables：查询中涉及到的表 Explain：SQL语句

--- EOF ---

阅读原文

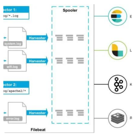
喜欢此内容的人还喜欢

聊聊sql优化的15个小技巧

架构师社区

企业级日志平台新秀Graylog，比ELK轻量多了

互联网后端架构



MySQL 索引失效的几种类型以及解决方式

互联网后端架构