4 款MySQL 調優工具，公司大神都在用！

方志朋 今天

以下文章來源方志朋的博客，回复”666“獲面試寶典

方志朋

號主為CSDN博客之星，博客訪問量突破一千萬，著有暢銷書《深入理解SpringCloud與微服務構建》。主要分享Java、後端架構等技術...

97篇原創內容

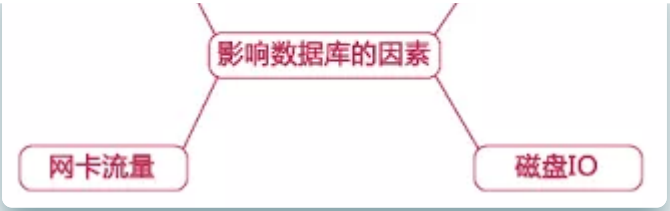
公眾號

來源：toutiao.com/a6691523026984370699

對於正在運行的mysql，性能如何，參數設置的是否合理，賬號設置的是否存在安全隱患，你是否了然於胸呢？

俗話說工欲善其事，必先利其器，定期對你的MYSQL數據庫進行一個體檢，是保證數據庫安全運行的重要手段，因為，好的工具是使你的工作效率倍增！

今天和大家分享幾個mysql 優化的工具，你可以使用它們對你的mysql進行一個體檢，生成awr報告，讓你從整體上把握你的數據庫的性能情況。



mysqltuner.pl

是mysql一個常用的數據庫性能診斷工具，主要檢查參數設置的合理性包括日誌文件、存儲引擎、安全建議及性能分析。針對潛在的問題，給出改進的建議。是mysql優化的好幫手。

在上一版本中，MySQLTuner支持MySQL / MariaDB / Percona Server的約300個指標。

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

1.1 下載

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

1.2 使用

```
[root@localhost ~]# ./mysqltuner.pl --socket /var/lib/mysql/mysql.sock
 >> MySQLTuner 1.7.4 - Major Hayden <major@mhtx.net>
 >> Bug reports, feature requests, and downloads at http://mysqltuner.com/
 >> Run with '--help' for additional options and output filtering
```

```
[--] Skipped version check for MySQLTuner script
Please enter your MySQL administrative login: root
Please enter your MySQL administrative password: [OK] Currently running supported MySQL version 5.7.23
[OK] Operating on 64-bit architecture
```

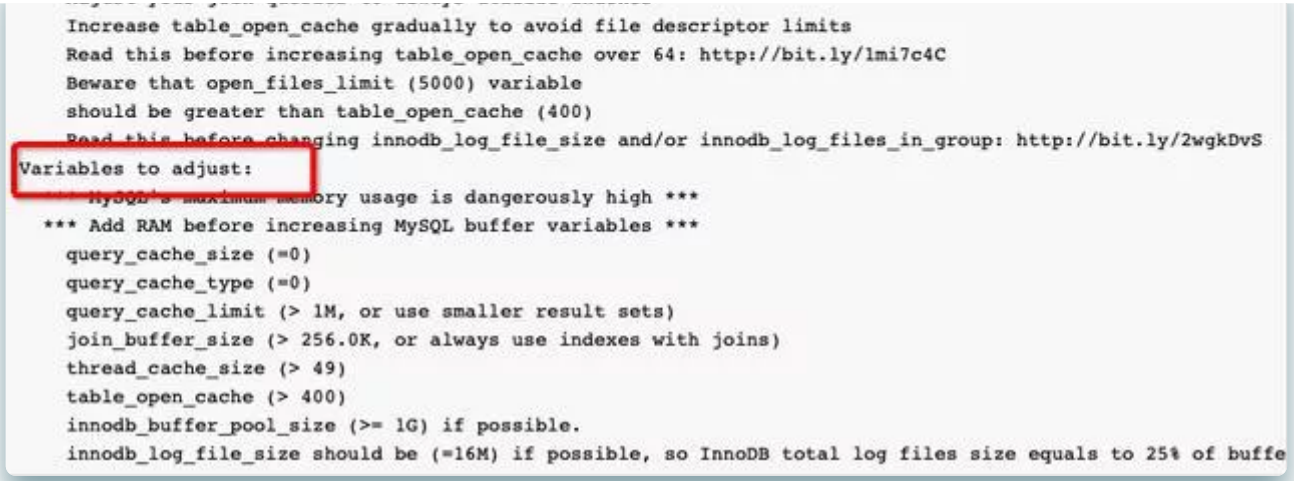
1.3、報告分析

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: 726.0M (26.80% 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% (55/2M)
[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
```

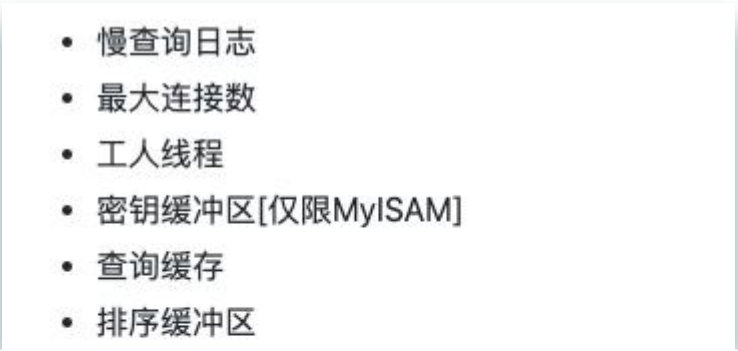


tuning-primer.sh

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：



- 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/tuning-primer.sh
```

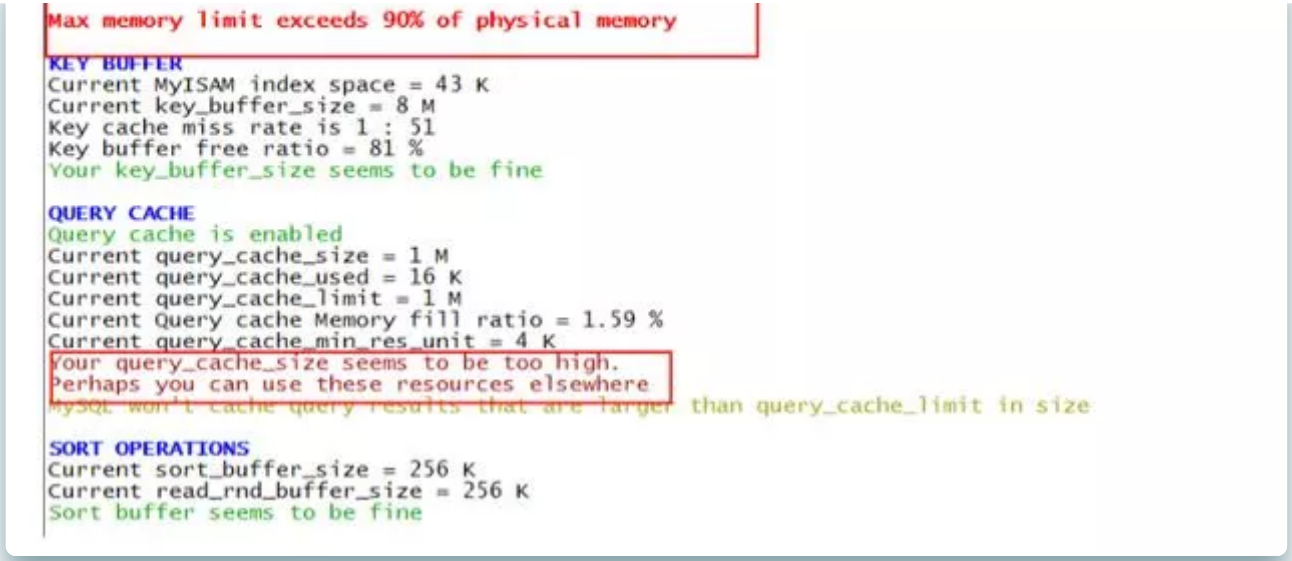
2.2 使用

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

 -- MYSQL PERFORMANCE TUNING PRIMER --
 - By: Matthew Montgomery -
```

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：



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/redhat/7/x86_64/percona-toolkit-3.0.13-re85ce15-el7-
[root@localhost ~]#yum install percona-toolkit-3.0.13-1.el7.x86_64.rpm
```

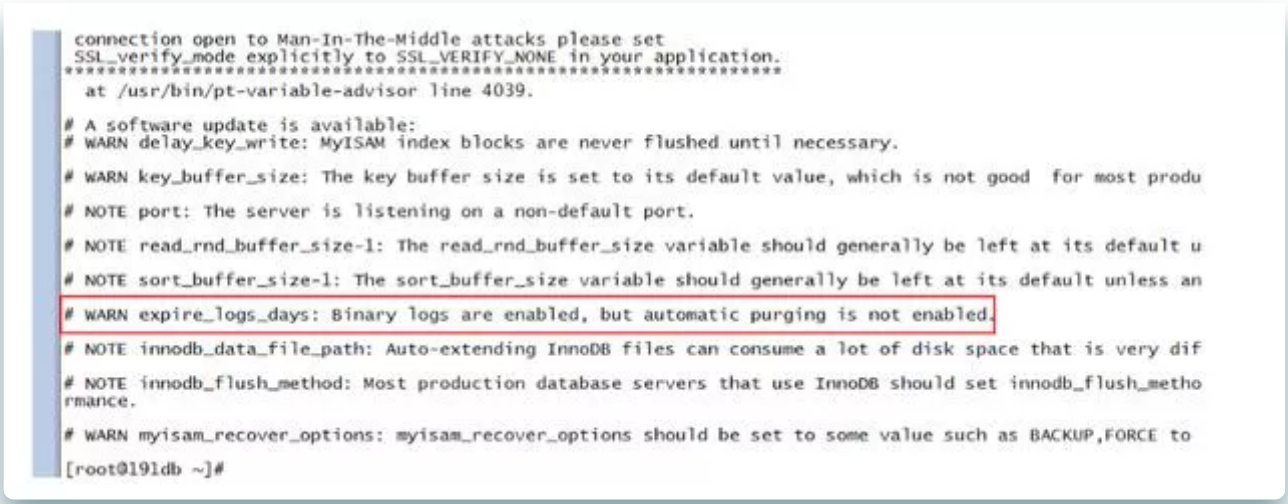
3.2 使用

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：



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 '2017-01-07 10:00:00'> > slow_report3.log
```

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-slow.log> slow_report4.log
```

5）针对某个用户的慢查询

```
pt-query-digest --filter '($event->{user} || "") =~ m/^root/i' /var/lib/mysql/slowtest-slow.log> slow_report5.log
```

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} || "") eq "yes")' /var/lib/mysql/slowtest-slow.log>
```

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

热门内容：

- Spring发布新成员：Spring GraphQL！高调出场的GraphQL能火起来了吗？

- 很哇塞的Java系列实战项目！
- 翻车！在项目中用了Arrays.asList、ArrayList的subList，被公开批评
- 60岁还在写代码的开发者，他的建议或许正是你现在焦虑的根源！尝试改变一下吧！

最近面试BAT，整理一份面试资料**《Java面试BAT通关手册》**，覆盖了Java核心技术、JVM、Java并发、SSM、微服务、数据库、数据结构等等。

获取方式：点“在看”，关注公众号并回复 666 领取，更多内容陆续奉上。

明天見(。·ω·。)/♡

喜歡此內容的人還喜歡

你的業務代碼，是不是都寫在了Activity裡?

架構師之路

16個Redis 常見使用場景

Java後端

免費的XShell替代品，又來一款國產良心工具....

Java技術前線