

UCloud 优惠价

十周年庆典钜惠

低至0.6折

快杰型云服务器
2核 4G 首年136元

立即购买

percona-toolkit工具集使用教程（pt工具）

作者：UStarGao 日期：2022-03-07 13:36:30 人气：512 栏目：MySQL 已收录

一、percona-toolkit工具集安装与介绍

percona-toolkit是一组高级命令行工具的集合，可以查看当前服务的摘要信息，磁盘检测，分析慢查询日志，查找重复索引，实现表同步等等。这个工具套件对DBA及运维人员着实不可多得。

官网地址：<https://www.percona.com/doc/percona-toolkit/3.0/index.html>

```
# 安装perl模块
yum install -y perl-CPAN perl-Time-HiRes

# RPM安装
wget https://downloads.percona.com/downloads/percona-toolkit/3.3.0/binary/redhat/7/x86_64/percona-toolkit-3.3.0-1.el7.x86_64.rpm
rpm -ivh percona-toolkit-3.3.0-1.el7.x86_64.rpm

# 如果出现报错依赖问题，可以按以下方式安装
yum install percona-toolkit-3.3.0-1.el7.x86_64.rpm -y
```

【注】工具安装目录在：/usr/bin下。我会在后续使用过程中，逐一给大家介绍~

```
[root@blogs-v2 ~]# ll /usr/bin/ |grep pt
-rwxr-xr-x 1 root root 41747 Jan 14 2021 pt-align
-rwxr-xr-x 1 root root 272712 Jan 14 2021 pt-archiver
-rwxr-xr-x 1 root root 3891 Nov 6 2016 ptaskset
-rwxr-xr-x 1 root root 171696 Jan 14 2021 pt-config-diff
-rwxr-xr-x 1 root root 168891 Jan 14 2021 pt-deadlock-logger
-rwxr-xr-x 1 root root 166450 Jan 14 2021 pt-diskstats
-rwxr-xr-x 1 root root 172014 Jan 14 2021 pt-duplicate-key-checker
-rwxr-xr-x 1 root root 50157 Jan 14 2021 pt-fifo-split
-rwxr-xr-x 1 root root 152724 Jan 14 2021 pt-find
-rwxr-xr-x 1 root root 67304 Jan 14 2021 pt-fingerprint
-rwxr-xr-x 1 root root 135870 Jan 14 2021 pt-fk-error-logger
-rwxr-xr-x 1 root root 225530 Jan 14 2021 pt-heartbeat
-rwxr-xr-x 1 root root 229128 Jan 14 2021 pt-index-usage
-rwxr-xr-x 1 root root 32405 Jan 14 2021 pt-ioprofile
-rwxr-xr-x 1 root root 13534872 Jan 14 2021 pt-k8s-debug-collector
-rwxr-xr-x 1 root root 257732 Jan 14 2021 pt-kill
-rwxr-xr-x 1 root root 21913 Jan 14 2021 pt-mext
-rwxr-xr-x 1 root root 12212984 Jan 14 2021 pt-mongodb-query-digest
-rwxr-xr-x 1 root root 12591128 Jan 14 2021 pt-mongodb-summary
-rwxr-xr-x 1 root root 108113 Jan 14 2021 pt-mysql-summary
-rwxr-xr-x 1 root root 438197 Jan 14 2021 pt-online-schema-change
-rwxr-xr-x 1 root root 6728024 Jan 14 2021 pt-pg-summary
-rwxr-xr-x 1 root root 24598 Jan 14 2021 pt-pmp
-rwxr-xr-x 1 root root 529350 Jan 14 2021 pt-query-digest
-rwxr-xr-x 1 root root 4795128 Jan 14 2021 pt-secure-collect
-rwxr-xr-x 1 root root 79159 Jan 14 2021 pt-show-grants
-rwxr-xr-x 1 root root 37784 Jan 14 2021 pt-sift
-rwxr-xr-x 1 root root 147869 Jan 14 2021 pt-slave-delay
-rwxr-xr-x 1 root root 133040 Jan 14 2021 pt-slave-find
-rwxr-xr-x 1 root root 186580 Jan 14 2021 pt-slave-restart
-rwxr-xr-x 1 root root 76186 Jan 14 2021 pt-stalk
-rwxr-xr-x 1 root root 90816 Jan 14 2021 pt-summary
-rwxr-xr-x 1 root root 461937 Jan 14 2021 pt-table-checksum
-rwxr-xr-x 1 root root 406401 Jan 14 2021 pt-table-sync
-rwxr-xr-x 1 root root 248658 Jan 14 2021 pt-table-usage
-rwxr-xr-x 1 root root 334032 Jan 14 2021 pt-upgrade
-rwxr-xr-x 1 root root 179330 Jan 14 2021 pt-variable-advisor
-rwxr-xr-x 1 root root 103460 Jan 14 2021 pt-visual-explain
```

二、percona-toolkit工具用法简介

2.1 常用工具介绍

(1) 慢查询日志分析统计

```
pt-query-digest /data/mysql/slow.log
```

在这里搜索...

Search

热门文章

关于网络排查工具MTR那些你必须了解的事情

Linux搭建GRE隧道教程

MySQL slow_log日志解读

开源资产管理系统chemex容器化部署

开源安全扫描工具OpenSCAP介绍

开源入侵检测系统OSSEC介绍

MySQL Binlog日志清理

Linux内网带宽压测工具-iperf3

MySQL多线程复制配置

最新文章

MySQL mysqldump备份之--set-gtid-purged=OFF参数影响

如何快速部署ChatGPT应用并绕开限制

Linux pstree命令树状图方式查看进程

Docker清理垃圾数据卷volumes

UCloud MySQL innodbackup物理备份还原到本地

Windows游戏服务器缺少.NET3.5环境依赖而诱发的时间戳转换问题

UCloud云数据库公网访问解决方案

Linux服务器OOM卡死解决方案-earlyoom

MongoDB切换主节点技巧

标签

备份未记录binlog 主从复制异常
--set-gtid-purged 参数
绕开ChatGPT限制
海外手机号解决方案 ChatGPT
Nginx 安全控制之IP白名单
进程树状图 pstree
Docker查看磁盘使用情况
Docker垃圾数据卷volumes清理
MySQL innodbackup物理备份还原
网络增强 智能网卡 DateTimeOffset
Windows游戏服务时间戳转换
Windows安装.NET3.5
全链路网络延迟探测 mysql-proxy
数据库代理 云NAT网关
数据库外网访问 云数据库 UDB
文件定位locate Linux后台运行程序
Linux用户管理 YAML 免密登录
dmesg stat查看文件修改时间 vim
dump Tab命令补全 lrzsz文件传输
网络出口查看 进程查看
Linux过滤配置 RPM包 yum源
自建从库 MySQL安装 安装MySQL
字符集

(2) 服务器摘要

```
pt-summary
```

(3) 服务器磁盘监测

```
pt-diskstats
```

(4) mysql服务状态摘要

```
pt-mysql-summary -- --user=root --password=root
```

2.2 pt-query-digest分析工具

pt-query-digest是用于分析mysql慢查询的一个工具，它可以分析binlog、General log、slowlog，也可以通过SHOWPROCESSLIST或者通过tcpdump抓取的MySQL协议数据来进行分析。可以把分析结果输出到文件中，分析过程是先对查询语句的条件进行参数化，然后对参数化以后的查询进行分组统计，统计出各查询的执行时间、次数、占比等，可以借助分析结果找出问题进行优化。

(1) pt-query-digest语法及重要选项

```
pt-query-digest [OPTIONS] [FILES] [DSN]
--create-review-table # 当使用--review参数把分析结果输出到表中时，如果没有表就自动创建
--create-history-table # 当使用--history参数把分析结果输出到表中时，如果没有表就自动创建
--filter # 对输入的慢查询按指定的字符串进行匹配过滤后再进行分析
--limit # 限制输出结果百分比或数量，默认值是20,即将最慢的20条语句输出，如果是50则按总响应时间占比从大到小排序，输出到总和达到50%位
--host # mysql服务器地址
--user # mysql用户名
--password # mysql用户密码
--history # 将分析结果保存到表中，分析结果比较详细，下次再使用--history时，如果存在相同的语句，且查询所在的时间区间和历史表中的不同，
--review # 将分析结果保存到表中，这个分析只是对查询条件进行参数化，一个类型的查询一条记录，比较简单。当下次使用--review时，如果存在相同的语句，且查询所在的时间区间和历史表中的不同，
--output # 分析结果输出类型，值可以是report(标准分析报告)、slowlog(MySQL slow log)、json、json-anon，一般使用report，以便于阅读
--since # 从什么时间开始分析，值为字符串，可以是指定的某个"yyyy-mm-dd [hh:mm:ss]"格式的时间点，也可以是简单的一个时间值：s(秒)、h(小时)、d(天)、w(周)、m(月)、y(年)
--until # 截止时间，配合--since可以分析一段时间内的慢查询
```

(2) 分析pt-query-digest输出结果

第一部分：总体统计结果

该工具执行日志分析的用户时间，系统时间，物理内存占用大小，虚拟内存占用大小

Current date:工具执行时间

Hostname：运行分析工具的主机名

Files: 被分析的文件名

Overall：语句总数量，unique唯一的语句数量（对查询条件进行参数化以后，总共有多少个不同的查询），QPS，并发数

Time range：查询执行的时间范围

total：总计、min：最小、max：最大、avg：平均、stddev：标准差

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

Exec time：语句执行时间

Lock time：锁占用时间

Rows sent：发送到客户端的行数

Rows examine：select语句扫描行数

Query size：查询的字符数

A software update is available:

```
# 220ms user time, 30ms system time, 28.96M rss, 241.93M vsz
# Current date: Wed Mar 9 10:08:26 2022
# Hostname: blogs-v2
# Files: slow.log
# Overall: 27 total, 10 unique, 0.00 QPS, 0.03x concurrency -----
# Time range: 2021-10-18T01:01:33 to 2021-10-18T10:34:37
# Attribute      total      min      max      avg      95%      stddev  median
# =====
# Exec time      1047s      4s       134s     39s      93s      32s      39s
# Lock time      219s      0        40s      8s       33s      13s      0
# Rows sent      8         0        1        0.30     0.99     0.45     0
# Rows examine   37.07k    0        37.04k   1.37k    1.96     6.91k    0
# Query size     3.99k     30       1.88k    146.04   158.58   371.38   31.70
```

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

```
# Profile
# Rank Query ID                Response time  Calls R/Call  V
# =====
# 1 0xDA5656F9115773A1A99AA0165670CE848 415.3448 39.7% 11 37.7586 4.81 ADMIN PREPARE
# 2 0xEF72850873E645A2D2C6E0A9A523B7E9 183.8802 17.6% 3 61.2934 8.97 SELECT ey_weapp
# 3 0xC73C2DDC687CE2874A03553B21C02CAD 134.3005 12.8% 1 134.3005 0.00 SELECT ey_weapp_spider_visit
# 4 0x680B6BE3ABEC0480A01E80CF6D64E13 100.5113 9.6% 1 100.5113 0.00 SELECT ey_language
# 5 0x1A972D87CAD4C8F480DC8C1AAC4E4D4 77.0878 7.4% 1 77.0878 0.00 SELECT ey_arctype
# 6 0x4243461F9B5B4A75DE54BE30A5D04E31 57.0082 5.4% 1 57.0082 0.00 SELECT ey_weapp_minicount_log
# 7 0x07890000813C4CC7111FD2D3F3B3B4EB 52.1332 5.0% 7 7.4476 0.96 ADMIN CLOSE STMT
# MISC 0xMISC                    27.1363 2.6% 2 13.5681 0.0 <2 ITEMS>
```



第三部分：每一种查询的详细统计结果

由下面查询的详细统计每个查询结果分析：
Query 6: 代表第6个查询
.....
Users：查询使用的用户信息
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

```
# Query 6: 0 QPS, 0x concurrency, ID 0x4243461F9B5B4A75DE54BE30A5D04E31 at byte 6166
# Scores: V/M = 0.00
# Time range: all events occurred at 2021-10-18T01:59:10
# Attribute pct total min max avg 95% stddev median
# =====
# Count 3 1
# Exec time 5 57s 57s 57s 57s 57s 0 57s
# Lock time 10 23s 23s 23s 23s 23s 0 23s
# Rows sent 12 1 1 1 1 1 0 1
# Rows examine 0 2 2 2 2 2 0 2
# Query size 2 99 99 99 99 99 0 99
# String:
# Hosts 172.17.0.1
# Users root
# Query_time distribution
# 1us
# 10us
# 100us
# 1ms
# 10ms
# 100ms
# 1s
# 10s+ #####
# Tables
# SHOW TABLE STATUS LIKE 'ey_weapp_minicount_log'\G
# SHOW CREATE TABLE `ey_weapp_minicount_log`\G
# EXPLAIN /*!50100 PARTITIONS*/
SELECT COUNT(*) AS tp_count FROM `ey_weapp_minicount_log` WHERE `update_time` > 1634493394 LIMIT 1\G
```



(3) pt-query-digest用法示例

```
# (1) 直接分析慢查询文件：
pt-query-digest slow.log > slow_report.log

# (2) 分析最近12小时内的查询：
pt-query-digest --since=12h slow.log > slow_report.log

# (3) 分析指定时间范围内的查询：
pt-query-digest slow.log --since '2021-05-16 09:30:00' --until '2021-05-16 10:00:00' > slow_report.log

# (4) 分析指含有select语句的慢查询
pt-query-digest --filter '{Sevent->{fingerprint}} =~ m/^select/i' slow.log > slow_report.log

# (5) 针对某个用户的慢查询
pt-query-digest --filter '{Sevent->{user}} || "" =~ m/^root/i' slow.log > slow_report.log

# (6) 查询所有所有的全表扫描或full join的慢查询
pt-query-digest --filter '({Sevent->{Full_scan}} || "") eq "yes" || ({Sevent->{Full_join}} || "") eq "yes"' slow.log > slow_report.log

# (7) 把查询保存到query_review表
pt-query-digest --user=root --password=123456 --review h=localhost,D=test,t=query_review--create-review-table slow.log
```

```
# (8) 把查询保存到query_history表
pt-query-digest --user=root --password=123456 --review h=localhost,D=test,t=query_history--create-review-table slow.log_0001

# (9) 通过tcpdump抓取mysql的tcp协议数据, 然后再分析
tcpdump -s 65535 -x -nn -q -tttt -i any -c 1000 port 3306 > mysql_tcp.txt
pt-query-digest --type tcpdump mysql_tcp.txt> slow_report.log

# (10) 分析binlog
mysqlbinlog mysql-bin.000100 > binlog.sql
pt-query-digest --type=binlog binlog.sql > binlog_report.log

# (11) 分析general log
pt-query-digest --type=genlog general.log > general_repor.log
```

2.3 pt-online-schema-change修改表结构工具

(1) 背景与痛点分析

在常见关系型数据库MySQL日常维护中，我们经常会遇到修改表结构的场景，针对较小的表修改表结构影响可以忽略不计，但当表规模达到百万，甚至过亿时，那么修改表结构影响的时间成本往往对于业务来说是不可接受的。下面让我们来看看修改表结构的影响有哪些：

- ◆ 在线修改大表的表结构执行时间往往不可预估，一般时间会非常长
- ◆ 由于修改表结构是表级锁，因此在修改表结构时，影响表写入操作
- ◆ 如果长时间的修改表结构，中途修改失败，由于修改表结构是一个事务，因此失败后会回滚表结构，在这个过程中表修改大表结构容易导致数据库CPU、IO等性能消耗，使MySQL服务器性能降低
- ◆ 在线修改大表结构容易导致主从延时，从而影响业务读取



(2) pt-online-schema-change工作原理

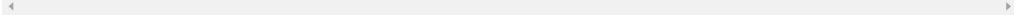
pt-online-schema-change是percona公司开发的percona-toolkit工具集中的一个，它可以实现在线修改表结构，并且能够做到业务无感知，大致原理如下：

- ◆ 首先pt-online-schema-change工具会新建一张与原表一模一样的表，表名一般是_new后缀
- ◆ 然后在这个新表执行更改字段操作
- ◆ 接着在源表上加三个触发器，DELETE/UPDATE/INSERT
- ◆ 从源表拷贝数据到新表，拷贝过程中源表通过触发器把新的DML操作也更新到新表中
- ◆ 最后rename源表到old表中，把新表rename为源表，默认最后删除源表

(3) 常用参数解读

```
--user=          # 连接mysql的用户名
--password=       # 连接mysql的密码
--host=          # 连接mysql的地址
P=3306           # 连接mysql的端口号
D=              # 连接mysql的库名
t=              # 连接mysql的表名
--alter          # 修改表结构的语句
--execute        # 执行修改表结构
--charset=utf8    # 使用utf8编码，避免中文乱码
--no-version-check # 不检查版本，在阿里云服务器中一般加入此参数，否则会报错
```

注1：其它参数可以使用pt-online-schema-change --help命令查看帮助文档。
注2：上述参数基本能够满足修改表结构场景。



作者：UStarGao

链接：<https://www.starcto.com/mysql/278.html>

来源：STARCTO

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。

UCloud云 平台 推荐

UCloud新用户专属注册连接

UCloud CDN超值特惠专场

UCloud全球云主机（UHost/VPs）大促页面

UCloud快杰云主机大促页面

10

标签： percona-toolkit工具集 pt-query-digest pt-summary pt-diskstats pt-mysql-summary

[查看上一篇](#) [查看下一篇](#)

随便看看

MySQL主从同步异常之从库误写数据	2021-07-21
Linux远程拷贝数据-限速和断点续传	2021-09-07
MySQL安全插件-密码复杂度+密码过期策略+登录失败处理	2021-07-03
MySQL Explain执行计划输出字段解读	2021-03-04
Windows远程桌面多个用户同时使用Chrome浏览器	2021-11-22