



PERCONA

Databases run better with Percona

PERCONA



Percona Toolkit na vida do DBA



Jean da Silva

- **Senior Database Performance Engineer**

O'Reilly – MongoDB Subject Matter Expert.

- MongoDB Fundamental – Interactive Labs.
- MongoDB Essentials – Video course.

Speaker :

- **Percona Live ONLINE 2021** – MongoDB Security Features.
- **Percona Live 2022** – PBM – The Backup Open Source Tool for MongoDB.

Percona Blog:

- [5 Changes You Should Know in MongoDB 7.0](#)
- [MongoDB Log and the Message “RSM Not Processing Response”](#)
- [Configuring Percona Backup for MongoDB in a Multi-Instances Environment](#)

[...]



Agenda

- O que é o Percona Toolkit.
- Como instalar.
- pt-summary.
- pt-mysql-summary.
- pt-stalk.
- pt-query-digest.
- pt-online-schema-change (pt-osc).
- Extras.



O que é o Percona Toolkit?



O Percona Toolkit **é um conjunto de ferramentas** avançadas, usadas para realizar tarefas complexas no MySQL, MongoDB, PostgreSQL e sistemas, que seriam difíceis de executar manualmente.

Essas ferramentas são **alternativas a scripts privados**, uma vez que foram desenvolvidas, formalmente testadas e completamente documentadas.

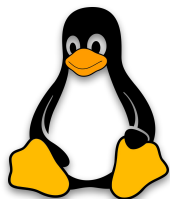
São totalmente autosuficientes, tornando a instalação rápida e fácil, sem a necessidade de instalação de bibliotecas adicionais.

O que é o Percona Toolkit?

- pt-align
- pt-archiver
- pt-config-diff
- pt-deadlock-logger
- pt-diskstats
- pt-duplicate-key-checker
- pt-eustack-resolver
- pt-fifo-split
- pt-find
- pt-fingerprint
- pt-fk-error-logger
- pt-galera-log-explainer
- pt-heartbeat
- pt-index-usage
- pt-ioprofile
- pt-k8s-debug-collector
- pt-kill
- pt-mext
- pt-mongodb-index-check
- pt-mongodb-query-digest

- pt-mongodb-summary
- pt-mysql-summary
- pt-online-schema-change
- pt-pg-summary
- pt-pmp
- pt-query-digest
- pt-secure-collect
- pt-show-grants
- pt-sift
- pt-slave-delay
- pt-slave-find
- pt-slave-restart
- pt-stalk
- pt-summary
- pt-table-checksum
- pt-table-sync
- pt-table-usage
- pt-upgrade
- pt-variable-advisor
- pt-visual-explain

O que é o Percona Toolkit?



- pt-summary
- pt-stalk
- pt-pmp
- pt-k8s-debug-collector
- pt-diskstats
- pt-ioprofile
- pt-eustack-resolver
- pt-fifo-split



- pt-mongodb-summary
- pt-mongodb-query-digest
- pt-mongodb-index-check



- pt-pg-summary
- pt-heartbeat



- pt-mysql-summary
- pt-query-digest
- pt-index-usage
- pt-mext
- pt-online-schema-change
- pt-config-diff
- pt-show-grants
- pt-variable-advisor
- pt-deadlock-logger
- pt-fk-error-logger
- pt-stalk
- pt-table-checksum
- pt-table-sync
- pt-archiver
- pt-slave-delay
- pt-slave-find
- pt-slave-restart
- pt-kill
- pt-upgrade
- pt-table-usage
- pt-visual-explain
- pt-galera-log-explainer

Como instalar

Sistema Operacional	EOL (Fim de Vida)	PT 3.X
RHEL (OL, CentOS) 8	01-May-2029	Ativo
RHEL (OL, CentOS) 9	01-May-2032	Ativo
Ubuntu 20.04 LTS	01-Apr-2025	Ativo
Ubuntu 22.04 LTS	01-Apr-2027	Ativo
Ubuntu 24.04 LTS	01-Apr-2029	Ativo
Debian 11	30-Jun-2026	Ativo
Debian 12	10-Jun-2028	Ativo

1. Via [Percona Repositories](#) e usando o gerenciador de pacote do seu sistema:

Se Debian ou Ubuntu:

```
sudo apt-get install percona-toolkit
```

Se RHEL or CentOS:

```
sudo yum install percona-toolkit
```

2. Baixando o pacote **.DEB** ou **.RPM** do [Download Center](#).

3. Ou baixando apenas a ferramenta que preferir via:

```
wget percona.com/get/<nome-da-ferramenta>
```



pt-summary

pt-summary

- **Objetivo:** Oferecer de forma útil o **status e a configuração de um servidor**. Não é uma ferramenta de Tuning ou diagnóstico.
 - Funciona **melhor quando executado como usuário root**, mas também pode ser usado sem privilégios, embora algumas informações possam não ser geradas.

- **Como executar:**

```
shell$> wget percona.com/get/pt-summary;  
shell$> chmod +x pt-summary;  
  
shell$> sudo ./pt-summary > "/tmp/pt-summary.out";
```

pt-summary

Percona Toolkit System Summary Report

```
Date | 2024-10-02 13:55:10 UTC (local TZ: -03 -0300)
Hostname | pc-oraclelinux8
Uptime | 3 days, 32 min, 3 users, load average: 0.00, 0.00, 0.00
System | Gigabyte Technology Co., Ltd.; X570S I AORUS PRO AX; v-CF (Desktop)
Service Tag | Default string
Platform | Linux
Release | Red Hat Enterprise Linux release 8.10 (Ootpa)
Kernel | 4.18.0-553.el8_10.x86_64
Architecture | CPU = 64-bit, OS = 64-bit
Threading | NPTL 2.28
SELinux | Disabled
Virtualized | No virtualization detected
```

Processor

```
Processors | physical = 1, cores = 8, virtual = 16, hyperthreading = yes
Speeds | 1x3617.126, 15x3800.000
Models | 16xAMD Ryzen 7 5800X 8-Core Processor
Caches | 16x512 KB
```

pt-summary

```
# Memory #####
Total | 15.5G
Free | 9.4G
Used | physical = 2.6G, swap allocated = 7.8G, swap used = 0.0, virtual = 2.6G
Shared | 45.2M
Buffers | 3.6G
Caches | 12.7G
Dirty | 124 kB
UsedRSS | 3.5G
Swappiness | 1
DirtyPolicy | 15, 5
DirtyStatus | 0, 0
Numa Nodes | 1
Numa Policy | default
Preferred Node | current
Node      Size      Free      CPUs
====      ==
node0     15913 MB    9603 MB    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

pt-summary

- Mounted Filesystems.
- Disk Schedulers And Queue Size.
- Disk Partitioning.
- Kernel Inode State.
- LVM Volume Groups.
- RAID Controller.
- Network Config.
- Interface Statistics.
- Network Devices.
- Network Connections.
- Top Processes.
- Notable Processes.
- Simplified vmstat.
- Memory management.



pt-mysql-summary

pt-mysql-summary

- **Objetivo:** Oferecer um resumo prático do status e da configuração **do MySQL**, permitindo uma visão geral de forma rápida e fácil.
Assim como **pt-summary**, não é uma ferramenta de diagnóstico nem tuning.

- **Como executar:**

```
shell$> wget percona.com/get/pt-mysql-summary;  
shell$> chmod +x pt-mysql-summary;  
  
shell$> sudo ./pt-mysql-summary --save-samples="/tmp/samples" -- --user=root  
--password=<mysql-root-password> > "/tmp/pt-mysql-summary.out";
```


pt-mysql-summary

```
# Percona Toolkit MySQL Summary Report #####
      System time | 2024-10-02 19:27:30 UTC (local TZ: -03 -0300)
# Instances #####
Port  Data Directory          Nice OOM Socket
=====
 8033 /home/jean/sandboxes/msb_8_0_33/data 0   1   /tmp/mysql_sandbox8033.sock
# MySQL Executable #####
      Path to executable | /home/jean/opt/mysql/8.0.33/bin/mysqld
      Has symbols | Yes
# Slave Hosts #####
No slaves found
# Report On Port 8033 #####
      User | root@localhost
      Time | 2024-10-02 16:27:30 (-03)
      Hostname | myfedora
      Version | 8.0.33 MySQL Community Server - GPL
      Built On | Linux x86_64
      Started | 2024-10-02 13:35 (up 02:52:13)
      Databases | 6
      Datadir | /home/jean/sandboxes/msb_8_0_33/data/
      Processes | 2 connected, 2 running
      Replication | Is not a slave, has 0 slaves connected
      Pidfile | /home/jean/sandboxes/msb_8_0_33/data/mysql_sandbox8033.pid (exists)
```

pt-mysql-summary

- Processlist.
- Status Counters.
- Table cache.
- Key Percona Server features.
- Percona XtraDB Cluster.
- Plugins.
- Schema.
- Noteworthy Technologies.
- InnoDB.
- MyISAM.
- Security.
- Roles.
- Encryption.
- Binary Logging.
- Noteworthy Variables.
- Configuration File.
- Memory management library.

pt-mysql-summary

```
/tmp/stamples$> ls -la
-rw-r--r--  1 root root    0 out  4 16:27 collect.err
-rw-r--r--  1 root root 13K out  4 16:27 mysql-status
-rw-r--r--  1 root root  60 out  4 16:27 mysql-databases
-rw-r--r--  1 root root 2,4K out  4 16:27 mysql-plugins
-rw-r--r--  1 root root    0 out  4 16:27 mysql-slave
-rw-r--r--  1 root root    0 out  4 16:27 ndb-status
-rw-r--r--  1 root root 4,4K out  4 16:27 innodb-status
-rw-r--r--  1 root root  582 out  4 16:27 mysql-processlist
-rw-r--r--  1 root root   10 out  4 16:27 mysql-users
-rw-r--r--  1 root root  495 out  4 16:27 mysql-roles
-rw-r--r--  1 root root    0 out  4 16:27 keyring-plugins
-rw-r--r--  1 root root  486 out  4 16:27 mysqld-instances
-rw-r--r--  1 root root    0 out  4 16:27 mysql-slave-hosts
-rw-r--r--  1 root root   39 out  4 16:27 mysqld-executables
-rw-r--r--  1 root root  149 out  4 16:27 mysql-master-logs
-rw-r--r--  1 root root   21 out  4 16:27 mysql-master-status
-rw-r--r--  1 root root 1,5K out  4 16:27 mysql-config-file
-rw-r--r--  1 root root 20K out  4 16:27 mysql-variables
drwxr-xr-x  2 jean jean  420 out  4 16:27 .
-rw-r--r--  1 root root 16K out  4 16:27 mysql-status-defer
```



pt-stalk

pt-stalk

- **Objetivo:** Coletar informações detalhadas **do MySQL e do servidor** quando ocorre problemas.
 - Funciona melhor quando executado como usuário root, mas também pode ser usado sem privilégios, embora algumas informações possam não ser geradas.
 - Certifique-se de que as ferramentas adicionais usadas pelo pt-stalk estejam instaladas no servidor, como os pacotes `sysstat`, `procp`s, `lsof`, `net-tools`, `numactl`, `pmap`.
- **Como executar:**

Atenção ⚠

Na verdade podemos utilizar o pt-stalk de duas formas:

1. Em *background*, aguardando um gatilho acontecer para então ativar a captura de dados.
2. Em *foreground*, mas sem a demanda de um gatilho específico, mas coletando os dados naquela execução.

pt-stalk

- Como executar em background:

Para executar precisamos dos gatilhos e definir que pt-stalk deva ser executado as a daemon, em background. Os gatilhos existentes são as opções `--function`, `--variable`, `--threshold` e `--cycles`.

- **--function**: A que observar.
 - O valor padrão monitora o *SHOW GLOBAL STATUS*, mas você também pode monitorar o *SHOW PROCESSLIST* ou especificar um arquivo com seu próprio código personalizado.
- **--variable**: A variável a ser comparada
 - O valor padrão monitora o contador *Threads_running*
- **--threshold**: Seta o valor máximo aceitável para o valor da **--variable**.
 - O valor padrão é 25.
- **--daemonize**: Faz com que a ferramenta seja executada em *background*, criando um processo em *fork* e registrando a saída conforme especificado na opção `--log`

pt-stalk

- Como executar em background:

```
shell$> wget percona.com/get/pt-stalk;  
shell$> chmod +x pt-stalk;
```

```
shell$> sudo ./pt-stalk --daemonize --variable=Threads_connected --threshold=1  
--log=/tmp/pt-stalk.log ---dest=/tmp/stalk_out/ -- --user=root --ask-pass --port=8033;
```

```
shell$> tail -f /tmp/pt-stalk.log  
2024_10_02_14_55_34 Starting ./pt-stalk --function=status --variable=Threads_connected  
--threshold=1 --match= --cycles=5 --interval=1 --iterations= --run-time=30 --sleep=300  
--dest=/tmp/stalk_out/ --prefix= --notify-by-email= --log=/tmp/pt-stalk.log  
--pid=/var/run/pt-stalk.pid --plugin=  
2024_10_02_14_55_34 Check results: status(Threads_connected)=2, matched=yes, cycles_true=1  
2024_10_02_14_55_35 Check results: status(Threads_connected)=2, matched=yes, cycles_true=2  
2024_10_02_14_55_36 Check results: status(Threads_connected)=2, matched=yes, cycles_true=3  
2024_10_02_14_55_37 Check results: status(Threads_connected)=2, matched=yes, cycles_true=4  
2024_10_02_14_55_38 Check results: status(Threads_connected)=2, matched=yes, cycles_true=5  
2024_10_02_14_55_38 Collect 1 triggered  
2024_10_02_14_55_38 Collect 1 PID 1038055  
2024_10_02_14_55_38 Collect 1 done  
2024_10_02_14_55_38 Sleeping 300 seconds after collect
```

pt-stalk

- Como executar em background:

Atenção ⚠

Para um uso eficaz do pt-stalk é recomendado uma boa condição de gatilho. Sendo assim:

- Sensível o suficiente para iniciar a coleta de dados quando um problema ocorre.
- Configurada alta o suficiente para evitar falsos positivos e coleta de dados quando o servidor está funcionando normalmente.

Caso contrário:

- Pode-se levar a falsos positivos.
- Coleta excessiva de dados.
- E impactar no desempenho do banco e no sistema de forma geral.

```
shell$> sudo ./pt-stalk --daemonize --variable=Threads_connected --threshold=100 --cycles=2
--iterations=10 --sleep=30 --log=/tmp/pt-stalk.log --dest=/tmp/stalk_out/ -- --user=root
--password=<mysql-root-password> --port=8033;

$ ps -ef | grep -i pt-stalk
root      1045104      2497  0 15:14 ?                00:00:00 bash ./pt-stalk --daemonize
--variable=Threads_connected --threshold=100 --cycles=2 --iterations=10 --sleep=30
--log=/tmp/pt-stalk.log ---dest=/tmp/stalk_out/ -- --user=root --password=msandbox --port=8033
```



```

-rw-r--r-- 1 root root 17115 out 4 15:18 2024_10_04_15_18_01-df
-rw-r--r-- 1 root root 120 out 4 15:18 2024_10_04_15_18_01-disk-space
-rw-r--r-- 1 root root 17556 out 4 15:18 2024_10_04_15_18_01-diskstats
-rw-r--r-- 1 root root 0 out 4 15:18 2024_10_04_15_18_01-dmesg
-rw-r--r-- 1 root root 4471 out 4 15:18 2024_10_04_15_18_01-innodbstatus1
-rw-r--r-- 1 root root 652638 out 4 15:18 2024_10_04_15_18_01-interrupts
-rw-r--r-- 1 root root 41713 out 4 15:18 2024_10_04_15_18_01-iostat
-rw-r--r-- 1 root root 2849 out 4 15:18 2024_10_04_15_18_01-iostat-overall
-rw-r--r-- 1 root root 924 out 4 15:18 2024_10_04_15_18_01-lock-waits
-rw-r--r-- 1 root root 1269 out 4 15:18 2024_10_04_15_18_01-log_error
-rw-r--r-- 1 root root 7285 out 4 15:18 2024_10_04_15_18_01-lsof
-rw-r--r-- 1 root root 34251 out 4 15:18 2024_10_04_15_18_01-meminfo
-rw-r--r-- 1 root root 66257 out 4 15:18 2024_10_04_15_18_01-mpstat
-rw-r--r-- 1 root root 4342 out 4 15:18 2024_10_04_15_18_01-mpstat-overall
-rw-r--r-- 1 root root 0 out 4 15:18 2024_10_04_15_18_01-mutex-status1
-rw-r--r-- 1 root root 7618589 out 4 15:18 2024_10_04_15_18_01-mysqldadmin
-rw-r--r-- 1 root root 82656 out 4 15:18 2024_10_04_15_18_01-netstat
-rw-r--r-- 1 root root 74508 out 4 15:18 2024_10_04_15_18_01-netstat_s
-rw-r--r-- 1 root root 3171 out 4 15:18 2024_10_04_15_18_01-numastat
-rw-r--r-- 1 root root 2033 out 4 15:18 2024_10_04_15_18_01-opentables1
-rw-r--r-- 1 root root 7824 out 4 15:18 2024_10_04_15_18_01-output
-rw-r--r-- 1 root root 17095 out 4 15:18 2024_10_04_15_18_01-pmap
-rw-r--r-- 1 root root 924 out 4 15:18 2024_10_04_15_18_01-prepared-statements
-rw-r--r-- 1 root root 25058 out 4 15:18 2024_10_04_15_18_01-processlist
-rw-r--r-- 1 root root 126914 out 4 15:18 2024_10_04_15_18_01-procstat
-rw-r--r-- 1 root root 79325 out 4 15:18 2024_10_04_15_18_01-procvmstat
-rw-r--r-- 1 root root 69515 out 4 15:18 2024_10_04_15_18_01-ps
-rw-r--r-- 1 root root 98868 out 4 15:18 2024_10_04_15_18_01-ps-locks-transactions
-rw-r--r-- 1 root root 1198155 out 4 15:18 2024_10_04_15_18_01-slabinfo
-rw-r--r-- 1 root root 8211 out 4 15:18 2024_10_04_15_18_01-slave-status
-rw-r--r-- 1 root root 56941 out 4 15:18 2024_10_04_15_18_01-sysctl
-rw-r--r-- 1 root root 929336 out 4 15:19 2024_10_04_15_18_01-top
-rw-r--r-- 1 root root 924 out 4 15:18 2024_10_04_15_18_01-transactions
-rw-r--r-- 1 root root 377 out 4 15:18 2024_10_04_15_18_01-trigger
-rw-r--r-- 1 root root 35795 out 4 15:18 2024_10_04_15_18_01-variables
-rw-r--r-- 1 root root 2876 out 4 15:18 2024_10_04_15_18_01-vmstat
-rw-r--r-- 1 root root 332 out 4 15:18 2024_10_04_15_18_01-vmstat-overall

```

Sistema e Recursos:

- dmesg, meminfo, sysctl, interrupts, procstat, procvmstat, numastat, slabinfo, disk-space, diskstats, iostat, iostat-overall, vmstat, vmstat-overall, df, mpstat, mpstat-overall

Rede:

- netstat, netstat_s

MySQL e InnoDB:

- innodbstatus1, mysqladmin, log_error, slave-status, processlist, ps-locks-transactions, opentables1, prepared-statements, mutex-status1, transactions, variables

Desempenho e Diagnóstico:

- ps, top, lsof, pmap, lock-waits

pt-stalk

- Como executar em foreground:

```
shell$> sudo ./pt-stalk --no-stalk --iterations=2 --sleep=30 --dest=/tmp/stalk_out/ --  
--user=root --password=<mysql-root-pass> --port=8033;
```

--no-stalk: Faz com que o **pt-stalk** inicie a coleta de dados **imediatamente**, sem esperar por um evento ou condição de gatilho.



pt-query-digest

pt-query-digest

- **Objetivo:** Ajudar a analisar queries através do **slow**, **general**, e **binary log** files.
 - Também pode ser usado para analisar queries do **SHOW PROCESSLIST** e do **tcpdump** ao monitorar o protocolo mysql.

Atenção ⚠

Embora seja possível utilizar o *pt-query-digest* com outros tipos de logs; o uso com o **slow-query log** é o **mais recomendado**.

Isso ocorre porque outras alternativas, como o **general log**, não contêm todos os atributos essenciais, como o **Query_time**. Da mesma forma, o uso com **tcpdump** requer tratamento prévio da saída para que as consultas possam ser analisadas corretamente. No caso do uso de **binlog** e, se você estiver utilizando **row-based replication**, a [ferramenta falhará](#).

pt-query-digest

- Pre-flight check:

```
mysql> SELECT @@global.long_query_time, @@global.slow_query_log, @@global.log_output,  
@@global.slow_query_log_file, @@global.min_examined_row_limit, @@global.log_slow_admin_statements\G;  
  
***** 1. row *****  
      @@global.long_query_time: 0.000000  
      @@global.slow_query_log: 1  
      @@global.log_output: FILE  
      @@global.slow_query_log_file: /var/log/mysql/slow.log  
      @@global.min_examined_row_limit: 0  
      @@global.log_slow_admin_statements: 1  
1 row in set (0.00 sec)
```

Atenção ⚠

Caso seu `log_output` seja **TABLE**, será necessário converter os dados para um arquivo de texto:

```
shell$> mysql -h<host> -u<mysql-user> -p<mysql-pass> -D mysql -s -r -e "SELECT CONCAT( '# Time:  
' , DATE_FORMAT(start_time, '%Y-%m-%dT%H:%i:%s'), 'Z', '\n', '# User@Host: ', user_host, '\n', '#  
Query_time: ', TIME_TO_SEC(query_time), ' Lock_time: ', TIME_TO_SEC(lock_time), ' Rows_sent: ',  
rows_sent, ' Rows_examined: ', rows_examined, '\nuse ', if(db='', 'mysql', db), '; \n', sql_text,  
';' ) FROM mysql.slow_log" > /tmp/mysql.slow.log
```

pt-query-digest

- **Pre-flight check:**

```
## slow-log file ##  
/home/jean/opt/mysql/8.0.33/bin/mysqld, Version: 8.0.33 (MySQL Community Server - GPL). started with:  
Tcp port: 8033  Unix socket: /tmp/mysql_sandbox8033.sock  
Time          Id Command  Argument  
# Time: 2024-10-02T21:12:39.394316Z  
# User@Host: root[root] @ localhost []  Id: 4188  
# Query_time: 0.025147  Lock_time: 0.000015 Rows_sent: 0  Rows_examined: 0 Thread_id: 4188 Errno: 0
```

pt-query-digest

- Como executar:

```
shell$> wget percona.com/get/pt-query-digest
shell$> chmod +x pt-query-digest;

shell$> ./pt-query-digest --limit=100% --since='2024-10-02T18:13:37'
--until='2024-10-02T18:16:47' /var/log/mysql/slow.log > /tmp/report.out;
```

193.9s user time, 730ms system time, 786.45M rss, 1006.46M vsz

Current date: Fri Oct 2 18:30:46 2024

Hostname: myfedora

Files: /var/log/mysql/slow.log

Overall: 2.46M total, 17 unique, 10.30k QPS, 1.85x concurrency -----

Time range: 2024-10-02T21:12:39 to 2024-10-02T21:16:38

# Attribute	total	min	max	avg	95%	stddev	median
# Exec time	443s	1us	221ms	179us	1ms	839us	31us
# Lock time	1s	0	7ms	0	1us	13us	0
# Rows sent	36.47M	0	100	15.54	97.36	34.53	0.99
# Rows examine	71.89M	0	200	30.63	192.76	61.50	0.99
# Bytes sent	4.61G	0	12.38k	1.96k	11.91k	4.21k	192.76
# Query size	313.82M	5	511.94k	133.69	234.30	6.25k	34.95
# Bytes receiv	278.43M	5	511.94k	118.61	202.40	6.25k	23.65
# Created tmp	0	0	0	0	0	0	0
# Created tmp	120.10k	0	1	0.05	0	0.22	0
# Errno	10.12k	0	1.26k	0.00	0	2.33	0
# Read first	0	0	0	0	0	0	0
# Read key	1.99M	0	38	0.85	0.99	0.37	0.99
# Read last	0	0	0	0	0	0	0
# Read next	46.91M	0	100	19.99	97.36	38.93	0
# Read prev	0	0	0	0	0	0	0
# Read rnd	0	0	0	0	0	0	0
# Read rnd nex	11.85M	0	101	5.05	0	21.21	0
# Sort merge p	0	0	0	0	0	0	0
# Sort range c	0	0	0	0	0	0	0
# Sort rows	23.46M	0	100	9.99	97.36	29.20	0
# Sort scan co	240.19k	0	1	0.10	0.99	0.30	

Profile

# Rank	Query ID	Response time		Calls	R/Call	V
#	====	=====		=====	=====	=
# 1	0xFFCA4D67EA0A788813031B8BBC3B329	307.4003	69.4%	122979	0.0025	0.00 COMMIT
# 2	0xE81D0B3DB4FB31BC558CAEF5F387E929	27.9057	6.3%	1229790	0.0000	0.00 SELECT sbtest?
# 3	0xF0C5AE75A52E847D737F39F04B198EF6	22.6722	5.1%	122979	0.0002	0.00 SELECT sbtest?
# 4	0x9934EF6887CC7A6384D1DEE77FA8D4C3	16.0683	3.6%	122979	0.0001	0.00 SELECT sbtest?
# 5	0xB2249CB854EE3C2AD30AD7E3079ABCE7	14.6793	3.3%	122979	0.0001	0.00 UPDATE sbtest?
# 6	0xA729E7889F57828D3821AE1F716D5205	12.4120	2.8%	122979	0.0001	0.00 SELECT sbtest?
# 7	0xFF7C69F51BBD3A736EEB1BFDCCF4EBCD	10.5842	2.4%	122979	0.0001	0.00 SELECT sbtest?
# 8	0x6C545CFB55365122F1256A27240AEFC7	7.3546	1.7%	122979	0.0001	0.00 INSERT sbtest?
# 9	0x8ED4A1775B8BF7E2A1EFA356C8F3EDBF	7.0046	1.6%	380	0.0184	0.01 INSERT sbtest?
# 10	0x410C2605CF6B250BE96B374065B13356	6.6783	1.5%	122979	0.0001	0.00 UPDATE sbtest?
# 11	0xDDBF88031795EC65EAB8A8A8BEEFF705	6.0440	1.4%	122979	0.0000	0.00 DELETE sbtest?
# 12	0xAC3DADD38A2CB41DF29DDC50968D05F1	2.0927	0.5%	10	0.2093	0.00 CREATE
# 13	0x8D589AFA4DFAEEED85FFF5AA78E5FF6A	1.9012	0.4%	122979	0.0000	0.00 BEGIN
# 14	0x6B829BBCBA666B7A48E0AFABFFAAA0A2	0.1338	0.0%	10	0.0134	0.00 CREATE TABLE sbtest? sbtest1
# 15	0xDA556F9115773A1A99AA0165670CE848	0.0191	0.0%	736	0.0000	0.00 ADMIN PREPARE
# 16	0x07890000813C4CC7111FD2D3F3B3B4EB	0.0016	0.0%	724	0.0000	0.00 ADMIN CLOSE STMT
# 17	0xEDBC971AEC392917AA353644DE4C4CB4	0.0000	0.0%	3	0.0000	0.00 ADMIN QUIT

Query 2: 9.32k QPS, 0.21x concurrency, ID 0xE81D0B3DB4FB31BC558CAEF5F387E929 at byte 620227053

This item is included in the report because it matches --limit.

Scores: V/M = 0.00

Time range: 2024-10-04T21:14:26 to 2024-10-04T21:16:38

#	Attribute	pct	total	min	max	avg	95%	stddev	median
#	Count	49	1229790						
#	Exec time	6	28s	6us	6ms	22us	57us	41us	12us
#	Lock time	42	439ms	0	427us	0	1us	0	0
#	Rows sent	3	1.17M	1	1	1	1	0	1
#	Rows examine	1	1.17M	1	1	1	1	0	1
#	Bytes sent	4	231.28M	197	199	197.20	192.76	0	192.76
#	Query size	13	42.34M	36	37	36.10	36.69	0.50	34.95
#	Bytes receive	10	28.15M	24	26	24.00	23.65	0.01	23.65
#	Created tmp	0	0	0	0	0	0	0	0
#	Created tmp	0	0	0	0	0	0	0	0
#	Errno	0	0	0	0	0	0	0	0
#	Read first	0	0	0	0	0	0	0	0
#	Read key	58	1.17M	1	1	1	1	0	1
#	Read last	0	0	0	0	0	0	0	0
#	Read next	0	0	0	0	0	0	0	0
#	Read prev	0	0	0	0	0	0	0	0
#	Read rnd	0	0	0	0	0	0	0	0
#	Read rnd nex	0	0	0	0	0	0	0	0
#	Sort merge p	0	0	0	0	0	0	0	0
#	Sort range c	0	0	0	0	0	0	0	0
#	Sort rows	0	0	0	0	0	0	0	0
#	Sort scan co	0	0	0	0	0	0	0	0
[...]									

```

[ ...]
# String:
# Databases      testdb
# End            2024-10-02... (4/0%), 2024-10-04... (3/0%)... 1220922 more
# Hosts          localhost
# Start          2024-10-02... (3/0%), 2024-10-04... (3/0%)... 1223580 more
# Users          root
# Query_time distribution
#   1us #####
#  10us #####
# 100us #
#   1ms #
#  10ms
# 100ms
#   1s
#  10s+
# Tables
#   SHOW TABLE STATUS FROM `testdb` LIKE 'sbtest7'\G
#   SHOW CREATE TABLE `testdb`.`sbtest7`\G
# EXPLAIN /*!50100 PARTITIONS*/
SELECT c FROM sbtest7 WHERE id=55313\G

```



pt-osc

pt-online-schema-change

pt-online-schema-change (*pt-osc*)

- **Objetivo:** Alterar tabelas *sem* bloqueios.
 - pt-osc altera a estrutura de uma tabela *sem* bloquear leituras ou escritas, e tem baixo impacto na operação concorrente do banco de dados.

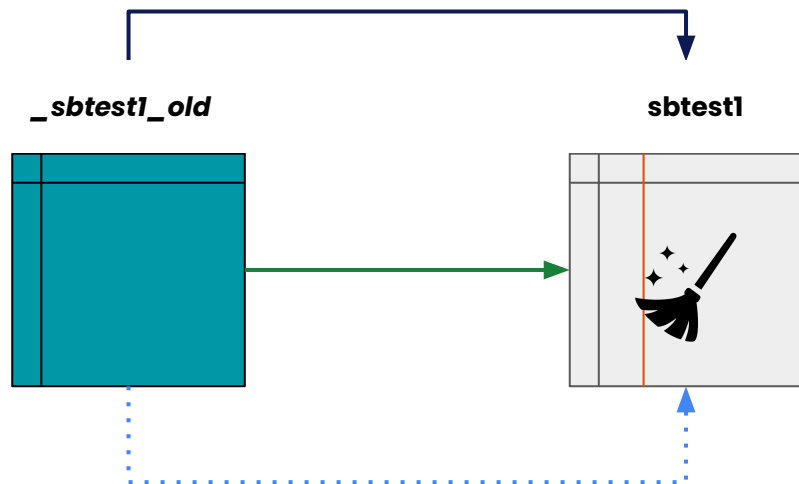
⚠ - Pode ser mais lento que executar um *ALTER TABLE* diretamente.

- **Como executar:**

```
shell$> wget percona.com/get/pt-online-schema-change;  
shell$> chmod +x pt-online-schema-change;  
  
shell$> pt-online-schema-change --alter "ADD COLUMN c1 INT" D=testdb,t=joinit --execute  
--host=127.0.0.1 --port=8033 --user=root --password=<mysql-root-password>
```

pt-online-schema-change (*pt-osc*)

- Como funciona:



1. *pt-osc* cria nova tabela

2. *pt-osc* modifica a nova tabela: **ADD COLUMN c1 INT**

3. *pt-osc* cria triggers na **_sbtest1_old** para capturar as mudanças:

- `pt_osc_testdb_sbtest1_del`
- `pt_osc_testdb_sbtest1_upd`
- `pt_osc_testdb_sbtest1_ins`

4. *pt-osc* copia os dados da **sbtest1** para **_sbtest1_new**.

5. *pt-osc* otimiza a **_sbtest1_new** via **ANALYZE TABLE**.

6. *pt-osc* renomeia a **sbtest1** para **_sbtest1_old** e **_sbtest1_new** para **sbtest1**

7. *pt-osc* remove **_sbtest1_old** e as triggers

pt-online-schema-change (*pt-osc*)

Pre-flight checks:

- **A tabela tem uma chave primária ou única?**
 - A ferramenta não funciona sem uma delas, pois precisa identificar as linhas individualmente para aplicar atualizações ou exclusões.
- **A tabela tem triggers?**
 - Se tiver, cancele a operação, a menos que você esteja no MySQL 5.7 e possa usar a opção `--preserve-triggers`.
- **Tem espaço suficiente em disco para rodar o pt-osc?**
 - Você vai precisar de pelo menos o tamanho atual do arquivo `*ibd`, ou o dobro disso se o log binário estiver ativado.
- **A tabela tem chaves estrangeiras?**
 - Se sim, escolha a opção correta com `--alter-foreign-keys-method`.

⚠ Sempre faça uma verificação completa com `--dry-run` antes de rodar. ⚠

pt-online-schema-change (*pt-osc*)

Atenção com seus triggers:

- Para versões do **MySQL anteriores à 5.7.2, o pt-osc não funcionará** na tabela se ela já tiver alguns triggers definidos.
 - Mas a partir da versão **5.7.2 do MySQL**, é possível definir múltiplos triggers para uma tabela.
 - Isso permite adicionar os triggers necessários como vimos anteriormente para o pt-osc.
- Caso esteja utilizando MySQL 5.7.2+ e procura preservar os triggers antigo desta tabela, utilize a opção **--preserve-triggers**.
 - Os triggers da tabela antiga são copiados **ANTES** de iniciar a cópia das linhas. Isso permite que os eventos sejam aplicados corretamente.

Atenção redobrada com as Foreign Keys:

- **FK complicam o uso da ferramenta e trazem riscos extras.**
- Dependendo do tamanho da tabela referenciada e da carga do servidor, **a forma como uma chave estrangeira (FK) é modificada para referenciar a nova tabela requer atenção**.
 - Diferentes parâmetros devem ser escolhidos usando a flag **--alter-foreign-keys-method**

pt-online-schema-change (*pt-osc*)

Tenha controle sobre o DDL:

O pt-osc possui parâmetros que podem ser utilizados para evitar sobrecargas que possam impactar a instância ou até mesmo replicação.

- **--max-load:** Definido para fazer com que o processo pause se um certo limite for atingido. A variável de status mais comum monitorada aqui é `Threads_running`, mas você pode usar outras variáveis relevantes.
- **--max-lag:** O número máximo de segundos de atraso que uma réplica pode ter antes que o pt-osc seja interrompido.
- **--max-flow-ctl:** Semelhante a **--max-lag**, verifica o tempo médio que o cluster **PXC** para por **flow-control** e interrompe a ferramenta se ultrapassar a porcentagem indicada.
- **--pause-file:** A execução será pausada enquanto o arquivo especificado por esse parâmetro existir.
 - *Exemplo:* se o parâmetro for `--pause-file=/tmp/pause-osc`, para pausar a execução, crie o arquivo com: `touch /tmp/pause-osc`. Para continuar a operação: `rm -f /tmp/pause-osc`.
 - A funcionalidade de pause não remove os triggers, o que significa que a nova tabela continua recebendo dados replicados enquanto estiver pausada. Apenas a cópia de dados é pausada.



Extras

Vale a pena mencionar

pt-table-checksum - Focada em encontrar diferenças/inconsistencia de dados na replicação.

+

pt-table-sync - Sincroniza os dados de forma eficiente entre tabelas.

```
shell$> pt-table-checksum --replicate=percona.checksums --ignore-databases mysql
h=localhost,u=checksum_user,p=checksum_password
      TS ERRORS  DIFFS      ROWS  CHUNKS SKIPPED    TIME TABLE
07-11T18:30:13      0      1       10        1        0     1.044 test.dummy
```

```
shell$> pt-table-sync --print --replicate=percona.checksums --sync-to-master
h=localhost,u=checksum_user,p=checksum_password
```

```
REPLACE INTO `test`.`dummy`(`id`, `name`) VALUES ('6', 'f') /*percona-toolkit src_db:test
src_tbl:dummy src_dsn:P=3306,h=192.168.0.130,p=...,u=checksum_user dst_db:test dst_tbl:dummy
dst_dsn:h=localhost,p=...,u=checksum_user lock:1 transaction:1 changing_src:percona.checksums
replicate:percona.checksums bidirectional:0 pid:24683 user:root host:slave*/;
```

```
REPLACE INTO `test`.`dummy`(`id`, `name`) VALUES ('7', 'g') /*percona-toolkit src_db:test
src_tbl:dummy src_dsn:P=3306,h=192.168.0.130,p=...,u=checksum_user dst_db:test dst_tbl:dummy
dst_dsn:h=localhost,p=...,u=checksum_user lock:1 transaction:1 changing_src:percona.checksums
replicate:percona.checksums bidirectional:0 pid:24683 user:root host:slave*/;
```

Vale a pena mencionar

pt-kill – Ajuda a encerrar consultas que correspondem a determinados critérios.

```
shell$> pt-kill --interval=10s --busy-time=2s --match-host=localhost --match-user=msandbox  
--match-command='Query|Execute' --victim=all --match-db=testdb --match-info='CREATE  
TEMPORARY|SELECT' --run-time=30 --print h=127.0.0.1,u=msandbox,p=msandbox,P=8033
```

```
## log ##
```

```
# 2021-05-04T04:09:58 KILL 57 (Query 9 sec) CREATE TEMPORARY TABLE test_TEMP (SELECT  
A.id,A.count from testBinlog A LEFT JOIN testBinlog B ON A.count=B.count)  
# 2021-05-04T04:10:08 KILL 57 (Query 19 sec) CREATE TEMPORARY TABLE test_TEMP (SELECT  
A.id,A.count from testBinlog A LEFT JOIN testBinlog B ON A.count=B.count)  
# 2021-05-04T04:10:18 KILL 57 (Query 29 sec) CREATE TEMPORARY TABLE test_TEMP (SELECT  
A.id,A.count from testBinlog A LEFT JOIN testBinlog B ON A.count=B.count)
```

PERCONA



Obrigado!