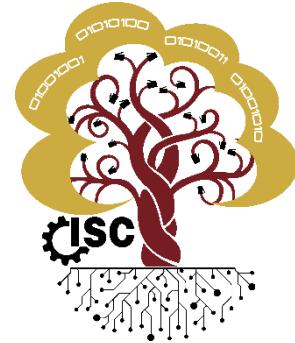




# INSTITUTO TECNOLÓGICO SUPERIOR DE



## INGENIERÍA EN SISTEMAS COMPUTACIONALES

### Administración de Base de Datos

Reporte de Práctica: MySQL SLAP

DOCENTE: MTI, ISC Salvador Acebedo Sandoval

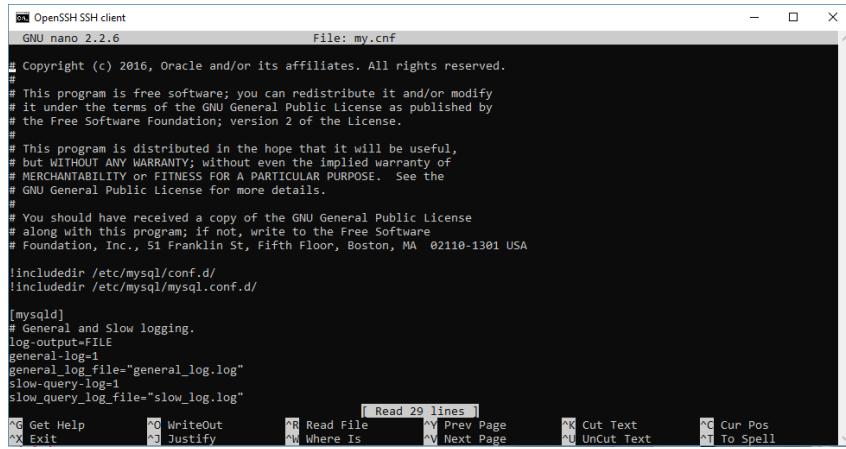
Edgar de la Cruz García 13070036 edg62@hotmail.com

## Objetivo

Conocer como es que se puede verificar cual es la capacidad de demanda del servidor por parte de usuarios para poder prevenir que un sistema falle, además de verificar como es que se activan las bitácoras en el servidor Debian y comprobar cuál es la información que se genera después de realizar las pruebas.

## Procedimiento

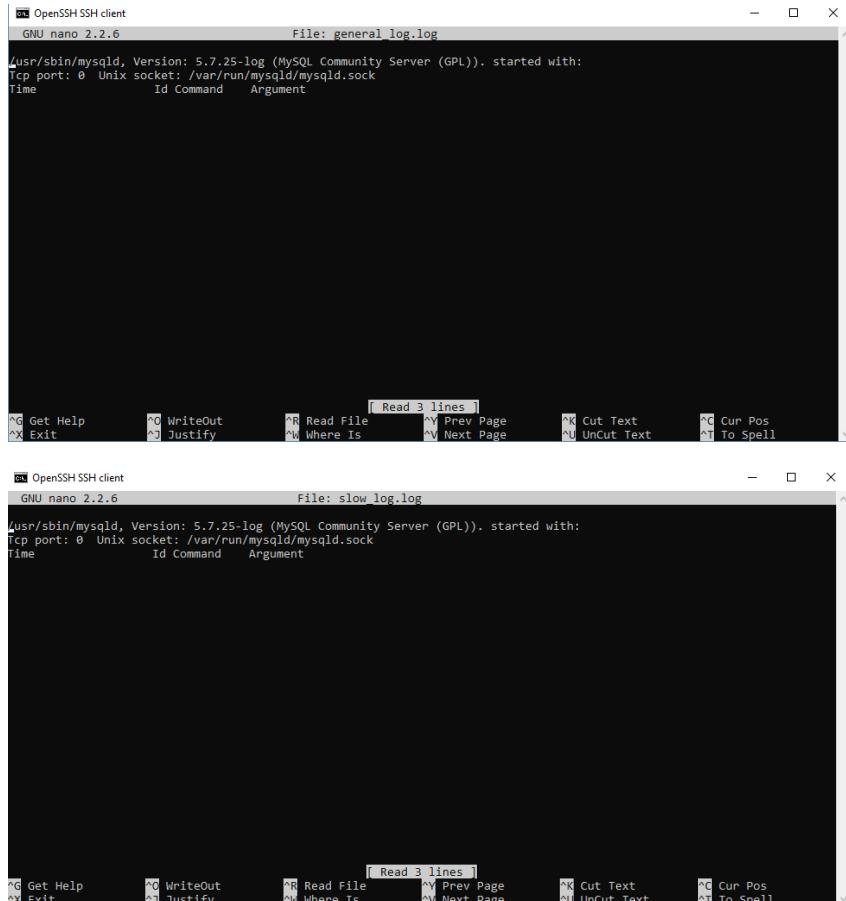
Para comenzar se activan las bitácoras, esto modificando el archivo my.cnf y activándolas de la siguiente manera:



The screenshot shows a terminal window titled "OpenSSH SSH client" with the file "my.cnf" open in the nano editor. The file contains MySQL configuration options, specifically for logging. The configuration includes sections for [mysqld] and [mysqld]. The [mysqld] section contains settings for log output, general log, and slow query logs. The terminal window has a standard nano interface with status bars at the top and bottom showing file statistics and command keys.

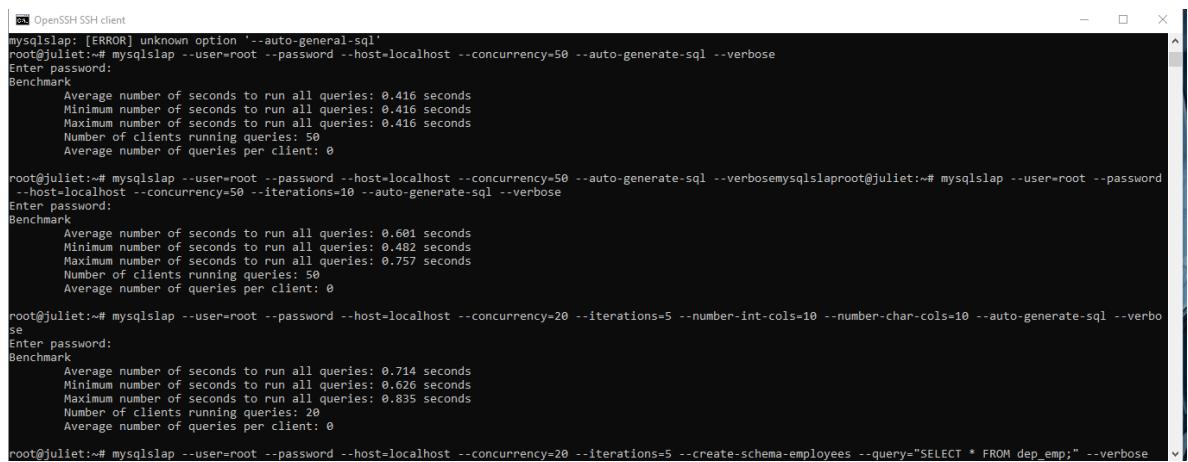
```
File: my.cnf
-
-  □ X
File: my.cnf
-
# Copyright (c) 2016, Oracle and/or its affiliates. All rights reserved.
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; version 2 of the License.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/
[mysqld]
# General and Slow logging.
log-output=FILE
general-log=1
general_log_file="general_log.log"
slow-query-log=1
slow_query_log_file="slow_log.log"
[mysqld]
# Get Help      ☐ WriteOut     ☐ Read File     ^Y Prev Page     ☐ Cut Text     ^C Cur Pos
☒ Exit        ☐ Justify      ☐ Where Is      ^V Next Page     ☐ Uncut Text    ^T To Spell
```

Una vez activadas las bitácoras y reiniciando el servidor podemos notar que los archivos generados se encuentran vacíos.



The image shows two separate terminal windows side-by-side. Both windows are titled "OpenSSH SSH client" and show the command "GNU nano 2.2.6". The top window is titled "File: general.log.log" and contains the text:  
"/usr/sbin/mysqld, Version: 5.7.25-log (MySQL Community Server (GPL)). started with:  
Tcp port: 0 Unix socket: /var/run/mysqld/mysqld.sock  
Time Id Command Argument". The bottom window is titled "File: slow.log.log" and contains the same text. Both windows have a menu bar at the top with options like "Get Help", "Exit", "WriteOut", "Read File", "Where Is", "Read 3 lines", "Prev Page", "Cut Text", "Cur Pos", "Justify", "UnCut Text", and "To Spell".

Se ejecutan diversas pruebas de estrés para observar como es que reacciona el servidor ante diversas pruebas.



The image shows a single terminal window titled "OpenSSH SSH client" with the command "GNU nano 2.2.6". It displays the output of several "mysqlslap" commands used for stress testing. The output includes error messages for unknown options and benchmark results for different test configurations. The benchmarks show average query times ranging from 0.416 to 0.714 seconds across 50 clients, with some iterations showing higher times due to concurrency. The terminal also shows the user entering a password for each command.

```
mysqlslap: [ERROR] unknown option '--auto-general-sql'
root@juliet:~# mysqlslap --user=root --password --host=localhost --concurrency=50 --auto-generate-sql --verbose
Enter password:
Benchmark
    Average number of seconds to run all queries: 0.416 seconds
    Minimum number of seconds to run all queries: 0.416 seconds
    Maximum number of seconds to run all queries: 0.416 seconds
    Number of clients running queries: 50
    Average number of queries per client: 0

root@juliet:~# mysqlslap --user=root --password --host=localhost --concurrency=50 --auto-generate-sql --verbose
mysqlslap: [ERROR] unknown option '--auto-general-sql'
root@juliet:~# mysqlslap --user=root --password --host=localhost --concurrency=50 --iterations=10 --auto-generate-sql --verbose
Enter password:
Benchmark
    Average number of seconds to run all queries: 0.601 seconds
    Minimum number of seconds to run all queries: 0.482 seconds
    Maximum number of seconds to run all queries: 0.757 seconds
    Number of clients running queries: 50
    Average number of queries per client: 0

root@juliet:~# mysqlslap --user=root --password --host=localhost --concurrency=20 --iterations=5 --number-int-cols=10 --number-char-cols=10 --auto-generate-sql --verbose
Enter password:
Benchmark
    Average number of seconds to run all queries: 0.714 seconds
    Minimum number of seconds to run all queries: 0.626 seconds
    Maximum number of seconds to run all queries: 0.835 seconds
    Number of clients running queries: 20
    Average number of queries per client: 0

root@juliet:~# mysqlslap --user=root --password --host=localhost --concurrency=20 --iterations=5 --create-schema=employees --query="SELECT * FROM dep_emp;" --verbose
```

## Resultado

Una vez realizadas las pruebas de estrés los archivos logs quedan de la siguiente manera, como se puede observar registran que es lo que se realizó en las pruebas.

The image shows a Windows desktop environment with two terminal windows open, both titled "OpenSSH SSH client" and running "GNU nano 2.2.6".

The top terminal window displays the contents of "general.log.log". The log entries show various MySQL queries and commands executed at approximately 08:36 a.m. on 27/03/2019. Key entries include:

```
Time           Id Command    Argument
2019-03-27T14:23:36.657047Z  2 Connect   root@localhost on using Socket
2019-03-27T14:23:36.677396Z  2 Query     DROP SCHEMA IF EXISTS `mysqlslap`  
2019-03-27T14:23:36.687492Z  2 Query     CREATE SCHEMA `mysqlslap`  
2019-03-27T14:23:36.702579Z  2 Init DB  mysqlslap  
2019-03-27T14:23:36.739938Z  2 Query     CREATE TABLE `t1` (intcol1 INT(32) ,charcol1 VARCHAR(128))  
2019-03-27T14:23:37.173364Z  2 Query     INSERT INTO t1 VALUES (1884289385,'mxvtVm9127qJNmeosGB8R92g2zj7Tl11TRD8  
2019-03-27T14:23:37.193828Z  2 Query     INSERT INTO t1 VALUES (822890675,'97RGH265mWzKsYfzWoSbgcIePz1bzsk13  
2019-03-27T14:23:37.210430Z  2 Query     INSERT INTO t1 VALUES (1388844878,'50w46158gielok1KoCvYf88g2BLADeg37jzgf3  
2019-03-27T14:23:37.218758Z  2 Query     INSERT INTO t1 VALUES (964445884,'DPhk7D1E6g4M0Qklioos01Ico0830D8Wu7685  
2019-03-27T14:23:37.227108Z  2 Query     INSERT INTO t1 VALUES (1586903199,'1wRHuW4HE31yv66UyIpcaAnr480d7Ymu3  
2019-03-27T14:23:37.245448Z  2 Query     INSERT INTO t1 VALUES (15869033002,'F1w4eg1L1isFxPkOc3nIx4rnwugT539kr5EX5  
2019-03-27T14:23:37.245814Z  2 Query     INSERT INTO t1 VALUES (1910858270,'ksnug3YyAhnW4DEJ1RK1fCA43eGkjJ2ijs5j5  
2019-03-27T14:23:37.260447Z  2 Query     INSERT INTO t1 VALUES (632979876,'F1w4eg1L1isFxPkOc3nIx4rnwugT539kr5EX5  
2019-03-27T14:23:37.268779Z  2 Query     INSERT INTO t1 VALUES (1348261729,'1wX2EnyclH7sDH1lxcILtoFQ2D0p0zjg448  
2019-03-27T14:23:37.277118Z  2 Query     INSERT INTO t1 VALUES (1804289383,'mxvtVm9127qJNmeosGB8R92g2zj7Tl11TRD8  
2019-03-27T14:23:37.293820Z  2 Query     INSERT INTO t1 VALUES (822890675,'97RGH265mWzKsYfzWoSbgcIePz1bzsk13  
2019-03-27T14:23:37.327105Z  2 Query     INSERT INTO t1 VALUES (1388844878,'50w46158gielok1KoCvYf88g2BLADeg37jzgf3  
2019-03-27T14:23:37.335456Z  2 Query     INSERT INTO t1 VALUES (964445884,'DPhk7D1E6g4M0Qklioos01Ico0830D8Wu7685  
2019-03-27T14:23:37.343776Z  2 Query     INSERT INTO t1 VALUES (1586903199,'1wRHuW4HE31yv66UyIpcaAnr480d7Ymu3  
2019-03-27T14:23:37.352118Z  2 Query     INSERT INTO t1 VALUES (962033002,'F1w4eg1L1isFxPkOc3nIx4rnwugT539kr5EX5  
2019-03-27T14:23:37.360455Z  2 Query     INSERT INTO t1 VALUES (1910858270,'ksnug3YyAhnW4DEJ1RK1fCA43eGkjJ2ijs5j5
```

The bottom terminal window displays the contents of "slow\_log.log". The log entries show various MySQL queries and commands executed at approximately 08:37 a.m. on 27/03/2019. Key entries include:

```
Time           Id Command    Argument
2019-03-27T14:23:37.173364Z  2 Connect   root@localhost on using Socket
2019-03-27T14:23:37.193828Z  2 Query     DROP SCHEMA IF EXISTS `mysqlslap`  
2019-03-27T14:23:37.210430Z  2 Query     CREATE SCHEMA `mysqlslap`  
2019-03-27T14:23:37.218758Z  2 Query     CREATE TABLE `t1` (intcol1 INT(32) ,charcol1 VARCHAR(128))  
2019-03-27T14:23:37.227108Z  2 Query     INSERT INTO t1 VALUES (1884289385,'mxvtVm9127qJNmeosGB8R92g2zj7Tl11TRD8  
2019-03-27T14:23:37.245448Z  2 Query     INSERT INTO t1 VALUES (822890675,'97RGH265mWzKsYfzWoSbgcIePz1bzsk13  
2019-03-27T14:23:37.245814Z  2 Query     INSERT INTO t1 VALUES (1388844878,'50w46158gielok1KoCvYf88g2BLADeg37jzgf3  
2019-03-27T14:23:37.260447Z  2 Query     INSERT INTO t1 VALUES (964445884,'DPhk7D1E6g4M0Qklioos01Ico0830D8Wu7685  
2019-03-27T14:23:37.268779Z  2 Query     INSERT INTO t1 VALUES (1348261729,'1wX2EnyclH7sDH1lxcILtoFQ2D0p0zjg448  
2019-03-27T14:23:37.277118Z  2 Query     INSERT INTO t1 VALUES (1804289383,'mxvtVm9127qJNmeosGB8R92g2zj7Tl11TRD8  
2019-03-27T14:23:37.293820Z  2 Query     INSERT INTO t1 VALUES (822890675,'97RGH265mWzKsYfzWoSbgcIePz1bzsk13  
2019-03-27T14:23:37.327105Z  2 Query     INSERT INTO t1 VALUES (1388844878,'50w46158gielok1KoCvYf88g2BLADeg37jzgf3  
2019-03-27T14:23:37.335456Z  2 Query     INSERT INTO t1 VALUES (964445884,'DPhk7D1E6g4M0Qklioos01Ico0830D8Wu7685  
2019-03-27T14:23:37.343776Z  2 Query     INSERT INTO t1 VALUES (1586903199,'1wRHuW4HE31yv66UyIpcaAnr480d7Ymu3  
2019-03-27T14:23:37.352118Z  2 Query     INSERT INTO t1 VALUES (962033002,'F1w4eg1L1isFxPkOc3nIx4rnwugT539kr5EX5  
2019-03-27T14:23:37.360455Z  2 Query     INSERT INTO t1 VALUES (1910858270,'ksnug3YyAhnW4DEJ1RK1fCA43eGkjJ2ijs5j5
```

## Conclusiones

Con base en los resultados obtenidos en estas pruebas se puede saber cual es la capacidad del servidor y en un caso real poder analizar los resultados y tomar medidas de prevención para evitar posibles fallos ante una gran cantidad de demandas.