mysqlsla v2 Guide

This document is an introductory guide and quick reference to using mysqlsla v2. mysqlsla v2 can be a simple script or a rather intricate instrument. Having nearly 50 **command line options** and recognizing over 100 MySQL log **meta-properties**, mysqlsla v2 has enough terms and jargon to choke an elephant.

However a complete and useful report is made with nothing more than two command line options: the type of log to parse and the log file itself (--log-type TYPE LOG).

Here is mysqlsla in one liners, the fast-track to getting started:

```
— Slow log: mysqlsla -lt slow slow.log

— General log: mysqlsla -lt general general.log

— Binary log: mysqlbinlog bin.log | mysqlsla -lt binary -
```

mysqlsla v2 Guide
Table of Contents

Installing

« Top

1. Download **mysqlsla-2.03.tar.gz**
2. tar xvfz mysqlsla-2.03.tar.gz
3. cd mysqlsla-2.03
4. perl Makefile.PL
5. make
6. make install

If all goes well, the installation procedure should copy mysqlsla to an appropriate bin directory and install its man page as well. Then, from the command line you should be able to run mysqlsla or man mysqlsla.

Alternatively, you can bypass the installer and simply wget http://hackmysql.com/scripts/mysqlsla-2.03, then chmod +x and run it.

Next Step: Filtering

« Top

One of mysqlsla's greatest features is its unrestricted filtering. A handful of other MySQL log parsing and filtering scripts exists but none come close to the extent of filtering that mysqlsla is capable.

There are two filters: the meta-property filter and the SQL statement filter (and grep if you wish to count that).

Meta-Property Filter

« Top

The meta-properties filter is set with the **meta-filter** option. A meta-property is a value or metric about a SQL statement: its execution time, how many rows it examined, etc. You probably already know a lot of meta-properties; the trick now is to learn how they are named according to mysqlsla.

The naming scheme is terse because there are a lot of meta-property names to manage. The list of all intenrally recognized meta-propert names is given in the document **mysqlsla v2 Filters**. They are organized according to MySQL log type and then according to meta-property type. Without comprehending that whole document you can simply scan through the list of meta-property names and find what you need.

SQL Statement Filter

« Top

The SQL statement filter is set with the **statement-filter** option. A **SQL statement type** is simply SELECT, UPDATE, DROP, INSERT—you know them already.

To filter out certain SQL statement types, set the **statement-filter** like: -TYPE,TYPE,TYPE. To allow only certain types, set it like: +TYPE,TYPE,TYPE. The only different is the leading - or + which says to make the filter negative (exclude the given TYPEs) or positive (allow only the given TYPEs).

Third Step: Sorting the Results

« Top

Remember those meta-property names used for the **meta-filter**? Almost any of them can be used as the value by which to sort the queries using the **sort** option. The exceptions were noted in the big list of meta-property names given in **mysqlsla v2 Filters**.

By default mysqlsla sorts the queries by `t_sum` (total query execution time) for slow and msl logs and `c_sum` (total number of times query appears in log) for all other log types. With the **sort** option you can sort the queries however you like.

Furthermore, by default mysqlsla only shows the top 10 queries. You can easily change that with the **top** option.

Reports

« Top

mysqlsla v2 has a number of **reports**: standard, time-all, print-unique, print-all, dump. One or more report can be made at once, or they can all be surpressed with the **silent** option (which is usually done when making **replays**).

If multiple reports are made, they are printed in the order given above. A **future feature** may introduce more

flexible report destinations but for now, with **replays**, it is not such a big problem.

standard

« Top

The standard report is the human-readable report which shows all the numbers and values calculated from the log. If no other report is specified, it is the default report.

mysqlsla automatically formats the standard report according to a report format depending on the log type being parsed. Therefore, the standard report for general logs is different from slow logs and binary logs, etc. mysqlsla has, internally, basic report formats for every log type, but a custom report format can be explicitly set by using the **report-format** option.

Read **mysqlsla v2 Reports** to learn how to create, modify and customize a standard report formats.

time-all

« Top

The time-all report times ALL queries from the log and reports the total time it took MySQL to execute them all. **WARNING**: you can destroy your database with this option because the queries are truly executed on the server. Therefore: a DROP DATABASE statement can truly drop the database!

print-unique

« Top

The print-unique report simple prints a sample of all unique queries, separating them with a blank line. This is useful for "exporting" SQL statements so that other scripts can read and anaylze them, such as the wonderful **mk-query-profiler** tool from the **Maatkit**. Something like the following actually works: mysqlsla -lt slow slow.log -R print-unique -mf "db=foo" -sf "+SELECT" | mk-query-profiler -separate -database foo

print-all

« Top

The print-all report prints ALL queries from the log, separating them wiht a blank line. This is like print-unique if you want instead to export every query, not just unique samples.

dump

« Top

The final report is dump which simply dumps mysqlsla's main data structures to screen. This is useful mostly for myself and those who wish to hack the mysqlsla internals. The first struct dumped is the unique queries hash, followed by the all users hash, and finally the grand totals hash.

Special Topics

« Top

The information provided below on these special topics is not meant to be an exhaustive resource, only a pointer in the right direction.

Microsecond Slow Logs

« Top

Certain versions of MySQL support microsecond values for long_query_time and therefore log microsecond values in the slow log. An outline of which versions of MySQL support this feature is given in the document **Microsecond Support for MySQL Slow Logs**.

mysqlsla v2 can parse either type of MySQL slow log: old-style with seconds resolution or new-style with microseconds resolution. The default standard report for slow logs is formatted in favor of the new-style with microsecond values. Therefore, if you are parsing old-style slow logs, you may notice a few decimal values.

Microslow (msl) Patched Slow Logs

« Top

If you are running a MySQL server patched with the **microslow (msl) patch** then you will be happy to know that mysqlsla v2 has full support for all the extra meta-properties that the patch introduces into the slow log. Making it work with mysqlsla is as simple as: --log-type msl.

The default standad report for msl even handles the case where some queries have the extra InnoDB values and others do not: if a query has them, then the report will print infromation about the 6 extra InnoDB values (IO r ops, Queue wait, etc.), otherwise it will print only the 8 basic meta-property values (QC hit, Full scan, etc.).

MySQL Proxy Logs (or User-Defined Logs)

« Top

It is becoming increasingly popular and easier to make your own MySQL logs, as **Giuseppe Maxia** demonstrated by writting a quick **Lua** script for **query and basic result logging** with **MySQL Proxy**. This is wonderful because the basic MySQL logs are, in my opinion, terribly planned: the general log is a parsing nightmare, the slow log does not have enough information (unless you are using the msl patch), the binary log is fraught with fluff, and all the logs are inconsistent in numerous ways.

So when you come to designing your own perfect MySQL log, mysqlsla is ready to parse and filter it as fully and probably more easily than the basic MySQL logs. The first step is defining the log format as discussed in the document **mysqlsla v2 UDL**.

The second step is creating your own standard report so that mysqlsla can turn the mass of computer-formatted data in the udl to a pretty little human-formatted report. This is discussed in the document **mysqlsla v2 Reports**.

Advancing

« Top

From here, the next step is reading all the mysqlsla command line options documented in **mysqlsla v2 Documentation**

naturally. These will give you ideas and hints about what else mysqlsla can do and refer you to the other mysqlsla documents which explain certain subjects in detail.

If after reading the myriad of documenation some feautre or usage is still unclear, you can **contact** me and I will help you. Though, please note that I will ignore basic questions like "how do I filter out only SELECT statements from the log?"—questions which are explained in multiple ways in multiple documents. (I have spent weeks writting these documents; please take at least a few hours to read them, too.)

mysqlsla v2 Guide was last updated July 9, 2008 for mysqlsla v2.00.