

MaxScale Release Notes

1.0 Beta

This document details the changes in version 1.0 since the release of the 0.7 alpha of the MaxScale product.

New Features

Complex Replication Structures

The MaxScale monitor module for Master/Slave replication is now able to correctly identify tree structured replication environments and route write statements to the master server at the root level of the tree. Isolated database instances are now also correctly identified as external to the replication tree.

Read/Write Splitter Enhancements

Support For Prepared Statements

Prepared statements are now correctly recognised by MaxScale, with the prepare stage being sent to all the eligible servers that could eventually run the statement. Statements are then executed on a single server.

Slave Failure Resilience

The Read/Write splitter can not be used to establish multiple connections to different slave servers. The read load will be distributed across these slaves and slave failure will be masked from the application as MaxScale will automatically failover to another slave when one fails.

Configurable Load Balancing Options

It is now possible to configure the criteria that the Read/Write Splitter uses for load balancing, the options are:

- The total number of connections to the servers, from this MaxScale instance
- The number of connections to the server for this particular MaxScale service
- The number of statements currently being executed on the server on behalf of this MaxScale instance
- Route statements to the slave that has the least replication lag

Replication Consistency

The Read/Write splitter may now be configured to exclude nodes that are currently showing a replication lag greater than a configurable threshold. The replication lag is measured using the MySQL Monitor module of MaxScale.

Alternatively it is possible to define that read operations should be routed to the slave that has the least measured replication lag.

Weighted Routing Options

The distribution of connections and statement across the set of nodes can be controlled by attaching arbitrary parameters to the servers and then configuring the router to use that parameter value as a weighting factor when deciding which of the valid servers to which to connect or route queries.

Several parameters may be used on each host and different routers may choose to use different parameters as the weighting parameter for that router. The use of weighting is optional, if no weighting parameter is given in the service definition then all eligible servers will have an equal distribution applied.

Server weighting is supported by both the Read/Write Splitter and the connection router.

MaxAdmin Client

A new administrative interface has been added that uses a MaxScale specific client application to interact with MaxScale to control and monitor the MaxScale activities. This client application may be used interactively or within scripts, passing commands to MaxScale via command line arguments. Command scripts are available, allowing command sets of commands to be stored in script files.

MaxAdmin also supports command history via libedit on those distributions that support the libedit library. This allows for the use of the up and down cursor keys or selection of previous commands and editing of lines using vi or emacs style editing commands.

Pacemaker Support

MaxScale now ships with an init.d script that is compatible with the use of Pacemaker and Heartbeat to provide for a highly available implementation of MaxScale. A tutorial on setting up MaxScale under Pacemaker control is included in the Documentation directory.

Filter API Enhancements

The filter API has now been enhanced to operate not just on downstream query filtering but also upstream result set filtering.

Enhanced and New Filters

Addition of new filters and enhancements to those existing filters that appeared in 0.7 of MaxScale.

Top Filter

A new filter to capture and log the longest running queries within a client session. The filter can be configured to capture a specific number of queries that take the longest time between the query being submitted to the database server and the first result being returned.

The queries captured can be defined using regular expressions to include and exclude queries that match these expressions. In addition the inclusion of a session may be based on the user name used to connect to the database or the source address of the client session.

Tee Filter

A filter to optionally duplicate requests received from the client and send them to other services within MaxScale. This allows a single statement sent by a client to be routed to multiple storage backends via MaxScale.

The queries duplicated can be defined using regular expressions to include and exclude queries that match these expressions. In addition the inclusion of a session may be based on the user name used to connect to the database or the source client session.

QLA and Regex Filter Improvements

These filters have been enhanced to provide for the inclusion of sessions by specifying the username used to connect to the database or the source of the client connection as a criteria to trigger the use of these filters for particular sessions connected to the MaxScale service.

Bug Fixes

A number of bug fixes have been applied between the 0.6 alpha and this alpha release. The table below lists the bugs that have been resolved. The details for each of these may be found in bugs.skysql.com.

ID	Summary
441	Possible failure to return a value in setipaddress
396	Build instruction suggest forcing install of RPM's
452	Make install copies the modules to an incorrect directory
450	Read/Write splitter does not balance load between multiple slaves
449	The router clientReply function does not handle GWBUF structures correctly

Packaging

Both RPM and Debian packages are available for MaxScale in addition to the tar based releases previously distributed we now provide

- CentOS/RedHat 5 RPM
- CentOS/RedHat 6 RPM
- Ubuntu 14.04 package