Configuring read-only database for expensive queries

Christopher Nadeau - 2023-08-31 - Comments (0) - Deskpro Legacy

For larger helpdesks, there are certain queries that can take a long time to run. For example, some reports can be quite intensive. To stop these intensive queries from slowing down the rest of the helpdesk you can configure a "read only" database to use instead.

MySQL Replication

To use this feature, you need to have a second database server that is a clone of your main, master database. You can achieve this through MySQL replication: http://dev.mysql.com/doc/refman/5.0/en/replication.html

Configuring the read-only database

In your /config.php file, locate the "Read Only Database" section.

```
##########################################################
# OPTIONAL : Read Only Database                          #
##########################################################
#                                                        #
# Configure whether a special database is used for       #
# particularly exprensive read queries including:        #
# - Reports                                              #
# - Searches and filters from the agent interace         #
# - Searches from the user interface                     #
#                                                        #
# For more advanced usage of this setting, read:         #
# https://support.deskpro.com/kb/articles/175            #
##########################################################

$DP_CONFIG['db_read'] = array();
$DP_CONFIG['db_read']['host'] = '';
$DP_CONFIG['db_read']['user'] = '';
$DP_CONFIG['db_read']['password'] = '';
$DP_CONFIG['db_read']['dbname'] = '';
```

Fill in the database details. That's it!

How it works

This feature works by simply changing the database that Deskpro sends certain read queries to. For example, when running a report, instead of executing the report against the

main database, it will open a new connection to your read database and execute it there instead.

Advanced Configuration

There are several different places where Deskpro can use a "read" database, and you can configure them each separately. As you saw above, you configured a read database by editing $DP_CONFIG['**db_read**'] variables in config.php. You can append a specific scope the "db_read" to have Deskpro use specific database config in specific places:

Location/Feature	Description	Deskpro Configuration Variable
Default	The default configuration used whenever a more specific configuration is not found.	db_read
Reports	The configuration used when running reports from the reports interface. If this configuration does not exist, the system will fallback to the Default configuration.	db_read_reports
Search	The configuration used when performing search-related features. You can create more specific configuration (described below), but this one will be used by default when no search-specific configuration exists. If this configuration itself does not exist, the system will fallback again to the Default configuration.	db_read_search
User Search	The configuration used in the user interface search, including the search bar at the top as well as searching for "related content" on the new ticket form. If this configuration does not exist, the system will fallback to the Search configuration, and then to the Default configuration.	db_read_search_searcher_content

Agent Filters	The configuration used to execute agent filters in the agent interface. This includes things like ticket filter numbers and listing, article lists, news lists, etc. If this configuration does not exist, the system will fallback to the Search configuration, and then to the Default configuration.	db_read_search_filter
Agent Filters (Specific)	These configurations are used for specific types of filtering. If these configurations do not exist, then the system will fallback to the Agent Filters configuration, then Search, then Default.	db_search_filter_tickets db_search_filter_people db_search_filter_articles db_search_filter_news db_search_filter_downloads db_search_filter_feedback

For examlpe, if we wanted to use a read database for **just** ticket filtering and nothing else, we would do two things;

1. Do **not** fill in the default db_read section in /config.php. By not specifying a default, we can be sure a read db is not used for other locaitons.
2. Create a **new** db_read_filter_tickets section like this:

```
$DP_CONFIG['db_read_search_filter_ticket'] = array();
$DP_CONFIG['db_read_search_filter_ticket']['host'] = 'localhost';
$DP_CONFIG['db_read_search_filter_ticket']['user'] = 'myuser';
$DP_CONFIG['db_read_search_filter_ticket']['password'] = 'mypass';
$DP_CONFIG['db_read_search_filter_ticket']['dbname'] = 'mydatabase';
```

Multiple Read Databases

You can also specify an **array** of configurations for the same scope, and Deskpro will choose a configuration at random. For example, say you have 3 databases you wanted to spread load across. You could create configuration like this which defines connection configurationf or all three, and Deskpro will just choose one randomly when a connection is required:

```
$DP_CONFIG['db_read_search_filter_ticket'] = array(
    array(
        'host'     => 'db.host_1',
        'user'     => 'myuser',
        'password' => 'mypass',
        'dbname'   => 'my_database'
    ),
    array(
        'host'     => 'db.host_2',
        'user'     => 'myuser',
        'password' => 'mypass',
        'dbname'   => 'my_database'
    ),
    array(
        'host'     => 'db.host_3',
        'user'     => 'myuser',
        'password' => 'mypass',
        'dbname'   => 'my_database'
    ),
);
```

Using PHP to dynamically select a database connection

Starting with build #321, you can now specify a PHP callback function that will be called when a read connection is requested. Specify it like this:

```
$DP_CONFIG['db_read_mapper'] = function($type, array $context = null) { /*...*/ }
```

$type will be a read type that is being requested. These types are currently supported:

- reports
- search
- search.searcher.content
- search.filter
- search.filter.(tickets|people|articles|news|downloads|feedback)

$context may be either null or an array of context variables. The context value will change based on which kind of connection is being requested. For example, in ticket searches $context['query'] will be the raw MySQL query that is waiting to be executed.

The function must *return* a string that represents the db_search_* config name. For example, if you wanted to use db_search_myconnection then you would return the string "myconnection". Returning any falsey value will result in Deskpro using the default logic for selecting a connection (as described above).

Here is an example that selects a new connection for all ticket searches that search on messages or subjects:

```
$DP_CONFIG['db_read_ticket_message_search'] = array(
        'host'     => 'db.example.com',
        'user'     => 'myuser',
        'password' => 'mypassword',
        'dbname'   => 'my_database'
);

$DP_CONFIG['db_read_mapper'] = function($type, array $context = null)
{
        if (
                $type == 'search.filter.tickets'
                && $context
                && !empty($context['query'])
                &&
preg_match('#(tickets_messages|tickets_search_message|tickets_search_
subject)#', $context['query'])
        ) {
                return 'ticket_message_search';
        }
};
```