Extending MySQL With The Component Infrastructure

Will MySQL Be Out Of Diskspace Soon?

Frédéric Descamps
Community Manager
Oracle MySQL
FOSDEM - February 2023
Who am I?

about.me/lefred
Frédéric Descamps

- @lefred
- MySQL Evangelist
- using MySQL since version 3.20
- devops believer
- living in 🇧🇪
- https://lefred.be
The Component Infrastructure

What is it?
The MySQL Component Infrastructure is a modular design for the MySQL Server that allows developers to extend the capabilities of the server in a variety of ways, such as adding support for new functions, performance_schema tables, variables, privileges...

The MySQL Component Infrastructure provides a set of services that components can use to interact with the rest of the server.

The best place find information is in the Component Services Inventory page: https://dev.mysql.com/doc/dev/mysql-server/latest/group__group__components__services__inventory.html

The Component Infrastructure is constantly evolving with new services. For example, there were 137 service definitions in MySQL 8.0.28, there are now 162 in 8.0.32!
MySQL Components

Why?
Components, why?

The component subsystem is designed to overcome some of the architectural issues of the plugin subsystem, namely:

- Plugins can only "talk" to the server and not with other plugins
- Plugins have access to the server symbols and can call them directly, i.e. no encapsulation
- There's no explicit set of dependencies of a plugin, thus it's hard to initialize them properly
- Plugins require a running server to operate.
Our Component

*what will it do, what does it need?*
What will it do?

Our new component, will:

- check if the user as the required privilege: `SENSITIVE_VARIABLES_OBSERVER` (8.0.29)

- read the value of some pre-defined variables where paths are specified

- create a Performance_Schema table with the path, the variable using it, the free space and total capacity of the storage related to that path
What does it need?

We need some services to access the required information but also to produce the output we want, messages and records in Performance_Schema.

The services we need are:

- `component_sys_variable_register`
- `log_builtins`
- `log_builtins_string`
- `mysql_thd_security_context`
- `mysql_security_context_options`
- `global_grants_check`
- `mysql_current_thread_reader`
- `mysql_runtime_error`
- `pfs_plugin_table_v1`
- `pfs_plugin_column_bigint_v1`
- `pfs_plugin_column_string_v2`
Which variables are we gonna check?

We will use the variables which define a path on the filesystem and we will put them in a vector of strings:
Which variables are we gonna check?

We will use the variables which define a path on the filesystem and we will put them in a vector of strings:

```cpp
std::vector<std::string> variables_to_parse {
    "log_bin_basename",
    "datadir",
    "tmpdir",
    "innodb_undo_directory",
    "innodb_data_home_dir",
    "innodb_log_group_home_dir",
    "innodb_temp_tablespaces_dir",
    "innodb_tmpdir",
    "innodb_redo_log_archive_dirs",
    "replica_load_tmpdir"
};
```
Which variables are we gonna check?

An improvement would be to also add this list into a MySQL variable.

```sql
SQL > select @@disksize.variables_to_parse;
+-----------------------------------------------+
| @@disksize.variables_to_parse                 |
+-----------------------------------------------+
| datadir;tmpdir;innodb_tmpdir;innodb_undo_directory |
+-----------------------------------------------+
1 row in set (0.0001 sec)
```

Using:

```php
mysql_service_component_sys_variable_register->register_variable()
```
Let's Code!

aren't we in a hacker conference?
Let's code!

We will add our component in the MySQL source tree inside the components directory:

```
mysql-server/components
  └── disksize
      ├── CMakeLists.txt
      ├── disksize.cc
      ├── disksize.h
      └── disksize_pfs.cc
```

CMakelists.txt  disksize.h  disksize.cc  disksize_pfs.cc
#define LOG_COMPONENT_TAG "disksize"

REQUIRES_SERVICE_PLACEHOLDER(log_builtins);
REQUIRES_SERVICE_PLACEHOLDER(log_builtins_string);

SERVICE_TYPE(log_builtins) * log_bi;
SERVICE_TYPE(log_builtins_string) * log_bs;

static mysql_service_status_t disksize_service_init()
{
    mysql_service_status_t result = 0;

    log_bi = mysql_service_log_builtins;
    log_bs = mysql_service_log_builtins_string;

    LogComponentErr(INFORMATION_LEVEL, ER_LOG_PRINTF_MSG, "initializing...");
    ...

...
bool have_required_privilege(void *opaque_thd) {
    // get the security context of the thread
    Security_context_handle ctx = nullptr;
    if (mysql_service_mysql_thd_security_context->get(opaque_thd, &ctx) || !ctx) {
        LogComponentErr(ERROR_LEVEL, ER_LOG_PRINTF_MSG,
                        "problem trying to get security context");
        return false;
    }

    if (mysql_service_global_grants_check->has_global_grant(
            ctx, PRIVILEGE_NAME, strlen(PRIVILEGE_NAME)))
        return true;

    return false;
}
Getting Global Variable's Value

```c
REQUIRES_SERVICE_PLACEHOLDER(global_grants_check);
...

const char *var_to_get = variables_to_parse.operator[](i).c_str();

if (mysql_service_component_sys_variable_register->get_variable(
    "mysql_server", var_to_get, (void **)&value, &value_length))
{
    sprintf(msgbuf, "Could not get value of variable [%s]", var_to_get);
    LogComponentErr(ERROR_LEVEL, ER_LOG_PRINTF_MSG, msgbuf);
    continue;
}
...
```
Demo!

Let's use our component...
Installing the Component

`sudo mysql --user=root --host=localhost`

```
[fred@dell ~] $ mysql
MySQL Shell 8.0.31

Copyright (c) 2016, 2022, Oracle and/or its affiliates. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type `\help` or `\?` for help; `\quit` to exit.
Creating a Classic session to `root@localhost`
Fetching global names for auto-completion... Press ^C to stop.
Your MySQL connection id is 8
Server version: 8.0.31 MySQL Community Server - GPL
No default schema selected; type `use <schema>` to set one.
```

MySQL Localhost 2023-01-02 11:39:39
SQL
```

```
SQL install component "file://component_disksize";
Query OK, 0 rows affected (0.0384 sec)
```

MySQL Localhost 2023-01-02 11:40:27
SQL
```
Installing the Component (2)

```
select * from performance_schema.error_log order by logged desc limit 2

1. row

LOGGED: 2023-01-02 21:42:26.936553
THREAD_ID: 9
PRIO: Note
ERROR_CODE: MY-011071
SUBSYSTEM: Server
  DATA: Component disksize reported: 'initializing...'

2. row

LOGGED: 2023-01-02 21:42:26.940168
THREAD_ID: 9
PRIO: Note
ERROR_CODE: MY-011071
SUBSYSTEM: Server
  DATA: Component disksize reported: 'PFS table has been registered successfully.'

2 rows in set (0.0106 sec)
```
### Getting Storage Information

```sql
select * from performance_schema.disks_size;
```

<table>
<thead>
<tr>
<th>DIR_NAME</th>
<th>RELATED_VARIABLE</th>
<th>FREE_SIZE</th>
<th>TOTAL_SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>/var/lib/mysql</td>
<td>log_bin_basename</td>
<td>212778520576</td>
<td>510389125120</td>
</tr>
<tr>
<td>/var/lib/mysql/</td>
<td>datadir</td>
<td>212778520576</td>
<td>510389125120</td>
</tr>
<tr>
<td>/var/tmp</td>
<td>tmpdir</td>
<td>212778520576</td>
<td>510389125120</td>
</tr>
<tr>
<td>.</td>
<td>innodb_undo_directory</td>
<td>212778520576</td>
<td>510389125120</td>
</tr>
<tr>
<td>.</td>
<td>innodb_log_group_home_dir</td>
<td>212778520576</td>
<td>510389125120</td>
</tr>
<tr>
<td>.</td>
<td>innodb_temp tablespaces_dir</td>
<td>212778520576</td>
<td>510389125120</td>
</tr>
<tr>
<td>/var/lib/mysql-redo-archive</td>
<td>innodb_redo_log_archives (backup1)</td>
<td>212778520576</td>
<td>510389125120</td>
</tr>
<tr>
<td>/var/tmp</td>
<td>replica_load_tmpdir</td>
<td>212778520576</td>
<td>510389125120</td>
</tr>
</tbody>
</table>

8 rows in set (0.0010 sec)
Getting Storage Information (2)

```
MySQL> select distinct * from (
    select dir_name, format_bytes(free_size) free_size,
    format_bytes(total_size) total_size,
    concat(round(((total_size-free_size)/total_size*100,2),"\"\") used
    from performance_schema.disks_size) order by 1;

+----------+---------+---------+------+
| dir_name | free_size | total_size | used |
|----------+----------+-----------+------|
| ./       | 198.16 GiB | 475.34 GiB | 58.31%|
| ./#innodb_temp/ | 198.16 GiB | 475.34 GiB | 58.31%|
| /var/lib/mysql | 198.16 GiB | 475.34 GiB | 58.31%|
| /var/lib/mysql-redo-archive/ | 198.16 GiB | 475.34 GiB | 58.31%|
| /var/lib/mysql/ | 198.16 GiB | 475.34 GiB | 58.31%|
| /var/tmp | 198.16 GiB | 475.34 GiB | 58.31%|

6 rows in set (0.0014 sec)
```
Privilege Error

```sql
show grants;

GRANTS for resto@localhost

| GRANT USAGE ON *.* TO 'resto'@'localhost'
| GRANT ALL PRIVILEGES ON 'docstore'.'restaurants' TO 'resto'@'localhost'

2 rows in set (0.0010 sec)
```

```sql
select * from performance_schema.disks_size;
ERROR: 1142: SELECT command denied to user 'resto'@'localhost' for table 'disks_size'
```
Misc FAQ

Some Info about MySQL Components
How can I known the loaded components?

```sql
select * from mysql.component;
```

<table>
<thead>
<tr>
<th>component_id</th>
<th>component_group_id</th>
<th>component_urn</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>file://component_query_attributes</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>file://component_uuid_v4</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>file://component_disksize</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>file://component_uuid_v7</td>
</tr>
</tbody>
</table>

4 rows in set (0.0007 sec)
In which folder are the available components installed?

```
select @@plugin_dir;
+---------------+
| @@plugin_dir   |
+---------------+
| /usr/lib64/mysql/plugin/ |
+--------------------------+
```
Usually, all the components start with `component_`:

```bash
$ ls component_*
component_audit_api_message_emit.so  component_mysqlbackup.so
component_disksize.so                 component_query_attributes.so
component_keyring_file.so            component_reference_cache.so
component_log_filter_dragnet.so      component_uuid_v4.so
component_log_sink_json.so           component_uuid_v7.so
component_log_sink_syseventlog.so    component_validate_password.so
```
Are loaded components still loaded after a restart of the server?

Yes, all components that are loaded will be loaded again when MySQL starts.
Are loaded components still loaded after a restart of the server?

Yes, all components that are loaded will be loaded again when MySQL starts.

Does MySQL start if a component was loaded when mysqld was stopped but the component file is removed?

Yes, MySQL will start and a message will be written in error log:

2022-02-16T13:47:54.394735Z 0 [ERROR] [MY-013129] [Server] A message intended for a client cannot be sent there as no client-session is attached. Therefore, we're sending the information to the error-log instead: MY-001126 - Can't open shared library '/usr/lib64/mysql/plugin/component_viruscan.so' (errno: 0 /usr/lib64/mysql/plugin/component_viruscan.so: cannot open shared object file: No such file or directory)
The Source Code

https://github.com/lefred/mysql-component-disksize
Now it's your turn!
Share your ❤️ to MySQL

#mysql

Join our slack channel!

bit.ly/mysql-slack
Questions ?
Resources & Credits

- https://dev.mysql.com/doc/dev/mysql-server/latest/group_group_components_services_inventory.html