MariaDB Server on macOS

On my lame attempts to build and use it on 10.13.6

Valerii Kravchuk, Principal Support Engineer, MariaDB
valerii.kravchuk@mariadb.com
What is this session about?

- Why bother to run MariaDB Server on macOS?
  - Because I can, I have a spare MacBook Air at hand, and I want to study and use **DTrace**, **Instruments** and other macOS specific tools, and test more (also on M1 eventually)
- Status of MariaDB Server on macOS
- Building recent MariaDB Server 10.1 … 10.6 from GitHub sources on macOS 10.13.6 (High Sierra), related issues and bugs reported
- My first lame attempts to use **DTrace** with MariaDB Server
- My first lame attempts to use **Instruments** with MariaDB Server
MariaDB Server on macOS - NOT SUPPORTED

- At least it is not supported by MariaDB Corporation or Foundation
- We do not provide binaries for macOS
- There is no macOS host in the buildbot
- Somebody cares, so there are different versions in MacPorts (that may work or not):

```
Valerii$ sudo port list | grep -i maria | grep server
mariadb-server       @5.5.68      databases/mariadb
mariadb-10.0-server  @10.0.38     databases/mariadb-10.0
mariadb-10.1-server  @10.1.48     databases/mariadb-10.1
mariadb-10.2-server  @10.2.41     databases/mariadb-10.2
mariadb-10.3-server  @10.3.32     databases/mariadb-10.3
mariadb-10.4-server  @10.4.22     databases/mariadb-10.4
mariadb-10.5-server  @10.5.13     databases/mariadb-10.5
mariadb-10.6-server  @10.6.5      databases/mariadb-10.6
mariadb-10.7-server  @10.7.1      databases/mariadb-10.7
```
MariaDB 10.6 Server on macOS from MacPorts...

- At least on my Air with macOS 10.13.6 there is a problem:
  Yuliyas-Air:~ Valerii$ sudo port install mariadb-10.6 mariadb-10.6-server
  ...
  ---> Attempting to fetch mariadb-10.6-10.6.5_1.darwin_17.x86_64.tbz2 from http://nue.de.packages.macports.org/mariadb-10.6
  ---> Building mariadb-10.6
  Error: Failed to build mariadb-10.6: command execution failed
  Error: See
  Error: Follow https://guide.macports.org/#project.tickets if you believe there is a bug.
  Error: Processing of port mariadb-10.6 failed
  ...
- But I still want to build and MariaDB server there...
MariaDB Server on macOS - Known Bugs

- I know I am not the only upstream user on macOS, as there are bug reports - 82 MDEVs matching now
- Who cares about them? 60 of the above are closed.
- I’ve reported the following while preparing for this talk:
  - MDEV-27402 - easy to workaround with a simple patch
  - MDEV-27403 - closed, surely affected 10.4+ badly when built with (unsupported) openssl 3. The fix may be even released already
  - MDEV-27579 - assigned, client, readline-related, may be workarounded with a patch
  - …
Building on macOS - prerequisites

- **This KB page** is a good start, but it’s far from being complete...
- **Install proper version of XCode** for your macOS version, from [here](https://developer.apple.com/xcode/)

  ```bash
  Yuliyas-Air:server Valerii$ clang -v
  Apple LLVM version 10.0.0 (clang-1000.10.44.4)
  ```

- **Install **MacPorts** (or Homebrew)**
- **Install git** and get the source code:

  ```bash
  git clone https://github.com/MariaDB/server.git
  git checkout 10.3
  git submodule update --init --recursive
  ```

- Openssl version 3 is installed by default (not supported until **10.8.1**), you’ll need 1.1 for MariaDB 10.4+:

  ```bash
  Yuliyas-Air:server Valerii$ sudo port list | grep openssl | head -4
  fbopenssl  @0.0.4  devel/fbopenssl
  openssl    @3        devel/openssl
  openssl110 @1.0.2u   devel/openssl110
  openssl111 @1.1.11   devel/openssl111
  ```
Building on macOS - MariaDB Server 10.1

- Basically build works without any issues, good old version, even though no longer supported
- Build out of source (mkdir buildtmp; cd buildtmp; …; rm -rf *;)
- Use as simple `cmake` command lines as possible. This usually works for my purposes:
  
  Yuliyas-Air:buildtmp Valerii$ cmake ..
  -DCMAKE_INSTALL_PREFIX=/Users/Valerii/dbs/maria10.1
  -DCMAKE_BUILD_TYPE=RelWithDebInfo -DENABLE_DTRACE=1

- Run MTR tests if you care about the usability of the results:
  
  Yuliyas-Air:mysql-test Valerii$ ./mtr --force
  ...

  MariaDB Version 10.1.49-MariaDB
  ...

  connect.json [ pass ] 204
  connect.part_file [ pass ] 292
  …
Building on macOS - MariaDB Server 10.2

- With this version I’ve hit my first problem during compilation, in the `mysql.cc` client code:

```c
…
[ 99%] Building CXX object client/CMakeFiles/mysql.dir/mysql.cc.o
/Users/Valerii/git/server/client/mysql.cc:1055:16: error: redefinition of
   '_hist_entry'
typedef struct _hist_entry {
   ^
/opt/local/include/editline/readline.h:57:16: note: previous definition is here
typedef struct _hist_entry {
   ^
1 error generated.
```

- It was easy to resolve with a lame patch, but I still reported it as **MDEV-27587**
With this version I’ve got my second problem, with MTR tests:

Yuliyas-Air:mysql-test Valerii$ ./mtr main.connect
Logging: ./mtr main.connect
...
MariaDB Version 10.3.33-MariaDB
...
main.connect [ fail ]
    Test ended at 2022-01-23 17:31:31

CURRENT_TEST: main.connect

Could not execute 'check-testcase' before testcase 'main.connect' (res: 1):
...
mysqltest: Start processing test commands from './include/check-testcase.test' ...
mysqltest: At line 19: query 'let $tmp= `SELECT '$tmp' = 'No such row'`' failed: 2013: Lost connection to MySQL server during query
Building on macOS - MariaDB Server 10.4+

- With these versions I was initially not able to get `mysql_install_db` to work without crashes (so MTR fails too)
- The solution was eventually to install openssl 1.1 from ports and make sure it is used:

```
Yuliyas-Air:buildtmp Valerii$ cmake ..
-DCMAKE_INSTALL_PREFIX=/Users/Valerii/dbs/maria10.6
-DCMAKE_BUILD_TYPE=RelWithDebInfo -DBUILD_CONFIG=mysql_release
-DFEATURE_SET=community -DWITH_EMBEDDED_SERVER=OFF
-DPLUGIN_TOKUDB=NO -DWITH_SSL=/opt/local/libexec/openssl11
-DENABLE_DTRACE=1
```

- 10.4 and 10.5 built cleanly (MTR tests problem aside), while for 10.6 I had to add `lame` fixes to workaround some problematic parts of the code when `__APPLE_CC__` is defined
DTrace on macOS - Basics

- Read about **DTrace** here, or there, or anywhere...
- It is there in macOS, ready to use:

```bash
Yuliyas-Air:maria10.6 Valerii$ dtrace
Usage: dtrace [-aACeFHlqSvWZ] [-arch i386|x86_64] [-b bufsz] [-c cmd]
[-D name[=def]]
[-x opt[=val]]

[-P provider [[ predicate ] action ]]
[-m [ provider: ] module [[ predicate ] action ]]
[-i probe-id [[ predicate ] action ]] [ args ... ]

predicate -> '//' D-expression '//'
    action -> '{' D-statements '}'
```

...
DTrace on macOS - using DTrace probes

- DTrace (USDT) probes in the code, or static tracepoints:

  
  Yuliyas-Air:maria10.6 Valerii$ sudo dtrace -l ./bin/mariadb | grep query-start

  ...
  2464 mysql73374 mariadb _ZL17wsrep_mysql_parseP3THDPCjP12Parser_state [wsrep_mysql_parse(THD*, char*, unsigned int, Parser_state*)] query-start
  2465 mysql73374 mariadb _Z16dispatch_command19enum_server_commandP3THDPCjb [dispatch_command(enum_server_command, THD*, char*, unsigned int, bool)] query-start

- How to use them:

  Yuliyas-Air:server Valerii$ grep -rni 'query-start' *


  ... 
  support-files/dtrace/query-rowops.d:30:mysql*:::query-start

  Yuliyas-Air:server Valerii$ cat support-files/dtrace/query-time.d
DTrace on macOS - query execution time

- Slightly shortened `query-time.d` from the above:

```c
#!/usr/sbin/dtrace -s
#pragma D option quiet
dtrace:::BEGIN {
    printf("%-20s %-20s %-40s %-9s\n", "Who", "Database", "Query", "Time(ms)");
}

mysql*:::query-start {
    self->query = copyinstr(arg0);
    self->connid = arg1;
    self->db = copyinstr(arg2);
    self->who = strjoin(copyinstr(arg3),strjoin("@", copyinstr(arg4)));
    self->querystart = timestamp;
}

mysql*:::query-done {
    printf("%-20s %-20s %-40s %-9d\n", self->who, self->db, self->query,
            (timestamp - self->querystart) / 1000000);
}
```

Yuliyas-Air:maria10.6 Valerii$ cat /tmp/query-time.d
DTrace on macOS - adding user probes

- Consider this lame example:

```c
#!/usr/sbin/dtrace -s
#pragma D option flowindent
pid$1::$2:entry {
    printf("Enter \%s\n", probefunc);
}

pid$1::$2:return {
    printf("Return from \%s\n", probefunc);
}
```

- Now let’s apply it to all functions in the PID of mariadb:

```bash
Yuliyas-Air:maria10.6 Valerii$ cat /tmp/userfunc.d
#!/usr/sbin/dtrace -s
#pragma D option flowindent
pid$1::$2:entry {
    printf("Enter %s\n", probefunc);
}

pid$1::$2:return {
    printf("Return from %s\n", probefunc);
}
```

```bash
Yuliyas-Air:maria10.6 Valerii$ sudo /tmp/userfunc.d 61198 '*' > /tmp/trace.txt
...
Failed to start process notifications for pid 61198 (19)
dtrace: script '/tmp/userfunc.d' matched 453035 probes
^C
```

```bash
Yuliyas-Air:maria10.6 Valerii$ ls -l /tmp/trace.txt
-rw-r--r--  1 Valerii  wheel  4162041 Jan 23 22:06 /tmp/trace.txt
```
Instruments on macOS - what you can do?

- **Instruments** looks impressive, DTrace-based GUI
MariaDB Server on macOS - Next Steps

- Studying **DTrace** and re-implementing/re-checking all Linux **bpftrace** scripts, at scale!
- Building Percona Toolkit and **sysbench** from sources
  
  Yuliyas-Air:sysbench Valerii$  `src/sysbench --version`
  
  sysbench 1.1.0-df89d34

- Studying of some MariaDB Server performance problems on macOS with better (?) instrumentation than ever
- Maybe working on proper fixes for MDEVs reported and new ones
- macOS upgrades and related experience. Maybe newer versions resolve problems I hit…
- Builds and tests on M1-based hardware, maybe one day…
- Blog posts pending. Stay tuned!
Thank you!

Questions and Answers?

Please, search and report bugs at:

- https://jira.mariadb.org