Compile and install MongoDB on Raspberry PI

Posted on **March 10, 2017** by **Koen** —

A 32 bit version of Raspbian is currently the most common OS that runs on a Raspberry PI. RPI 3 comes equipped with a 64 bit CPU, however at the moment only experimental 64 bit versions of Raspbian may be found. There is, however a 64 bit ARM version of SUSE Linux available for RPI 3 and probably in the near future 64 bit platforms for RPI 3 will become more common.

The highest MongoDB version that will run on 32 bit is v3.2. As of MongoDB v3.4 only 64 bit versions are supported. MongoDB v3.2 is the latest version that you will be able to run on a 32 bit Raspbian. In addition, you will not be able to run MongoDB with the WiredTiger storage engine since it only works on 64 bit versions. You will have to start the DB with the MMAPv1 storage engine. Also MMAPv1 has a limit of 2GB of storage, at least on ARM32. Once your DB hits that limit, you're out of luck.

Below you will find steps for building MongoDB for both ARM32 and ARM64 (aarch64).

Building MongoDB on ARM32

The steps below describe how you can compile MongoDB 3.2 on your Raspberry PI with 32 bit ARM architecture.

```
$ uname -a
Linux mypi 4.4.50-v7+ #970 SMP Mon Feb 20 19:18:29 GMT 2017 armv7l GN
```

1) Log in as a regular user.

For instance "pi" user and cd to home directory.

```
$ cd
```

2) Update/upgrade your system.

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install wget
```

3) Download and unpack mongodb source code.

```
$ mkdir install
$ cd install
$ wget https://fastdl.mongodb.org/src/mongodb-src-r3.2.12.tar.gz
$ tar xvf mongodb-src-r3.2.12.tar.gz
$ cd mongodb-src-r3.2.12
```

4) Check install steps in docs/building.md and verify prerequisites.

Note that scons will be installed in the next step, in case you don't have it already.

```
$ cat docs/building.md
$ gcc --version
$ python --version
$ scons --version
```

5) Install dependencies.

```
$ sudo aptitude install scons build-essential
$ sudo aptitude install libboost-filesystem-dev libboost-program-opti
$ sudo aptitude install python-pymongo
```

6) Temporarily increase swap space.

You may run into compilation and/or linking issues due to insufficient swap space. You can temporarily increase it with the steps below. Don't forget to delete the swap file after you reboot.

```
sudo dd if=/dev/zero of=/mytempswapfile bs=1024 count=524288
sudo chmod 0600 /mytempswapfile
sudo mkswap /mytempswapfile
sudo swapon /mytempswapfile
```

7) Generate additional sources.

```
$ cd src/third_party/mozjs-38/
$ ./get_sources.sh
$ ./gen-config.sh arm linux
$ cd -
```

Note: if you skip these steps you may encounter the following compilation error:

```
Compiling build/opt/mongo/scripting/mozjs/base.o
In file included from :0:0:
/usr/include/stdc-predef.h:59:1: fatal error: js-config.h: No such file or
directory
#endif
^
compilation terminated.
scons: *** [build/opt/mongo/scripting/mozjs/base.o] Error 1
```

```
scons: building terminated because of errors.
```

8) Compile and build.

Note: this could take a long time.

While you certainly could run “`scons all`” you will run into several challenges. First of all, WiredTiger will not work on a 32 bit architecture so you will encounter some compilation challenges with it. It is recommended to simply exclude the WiredTiger storage engine altogether. That means you will run MongoDB with the MMAPv1 storage engine only. Second, if you just want to run mongod/mongos and mongo you don’t need to compile all the tests, which will take a very long time to compile and link. Third, compiling everything will generate somewhere in the neighbourhood of 25GB of build files, while just mongo, mongod and mongos will take up around 4GB.

Probably all you need is the following:

```
$ scons core --wiredtiger=off --mmapv1=on
```

Or if you’re not going to use mongos you could even compile as follows:

```
$ scons mongo mongod --wiredtiger=off --mmapv1=on
```

Note that you need to explicitly specify the mmapv1 option as shown above, otherwise this storage engine will not get compiled into the mongod binary.

9) Reduce binaries file size.

You will notice that the binaries that are generated under ./build/opt/mongo are quite large:

```
-rwxr-xr-x 1 pi pi 156M Mar 7 13:26 mongo
-rwxr-xr-x 1 pi pi 429M Mar 9 14:36 mongod
-rwxr-xr-x 1 pi pi 217M Mar 9 16:52 mongos
```

That is because they contain debugging information. You can remove this with the strip command, for instance

```
$ strip -s mongo
```

The resulting files are now much smaller:

```
-rwxr-xr-x 1 pi pi 14M Mar 9 20:32 mongo
-rwxr-xr-x 1 pi pi 27M Mar 9 20:32 mongod
-rwxr-xr-x 1 pi pi 13M Mar 9 20:33 mongos
```

10) Copy binaries.

Finally copy the binaries to their bin location:

```
$ cd build/opt/mongo
$ sudo cp mongo mongod mongos /usr/local/bin/
```

Building MongoDB on aarch64 (ARM64)

The steps below describe how you can compile MongoDB 3.4 on your Raspberry PI with 64 bit ARM architecture.

```
$ uname -a
Linux mypi 4.4.57-18.3-default #1 SMP Thu Mar 30 06:39:47 UTC 2017 (3
```

If you want to install MongoDB 3.4 and higher on your Raspberry PI 3, you will first have to install 64-bit openSUSE Leap (JeOS Image). At the time of this writing, the latest version is Leap 42.2. It works great but you may run into wlan issues that can luckily be resolved. Once you have the 64 bit SUSE platform running on your RPI 3 you will have to use zypper to install gcc5 or gcc6 and scons. Then download the mongodb source code for 3.4 or higher. You can compile with “scons mongod –wiredtiger=on –mmapv1=off –disable-warnings-as-errors”. You may run into a build problem for the WiredTiger engine however the fix is rather simple. I added 2 CFLAGS lines in `mongodb-src-r3.4.3/src/third_party/wiredtiger/SConscript`:

```
elif env.TargetOSIs('linux'):
    env.Append(CFLAGS=["-march=armv8-a+crc"])
    env.Append(CFLAGS=["-mtune=generic"])
    env.Append(CPPPATH=["build_linux"])
    env.Append(CPPDEFINES=["_GNU_SOURCE"])
```

Once the build is done, don’t forget to reduce the binary files with the `strip` command.

Note that you can’t compile MongoDB 3.2 on an aarch64 architecture, you need at least version 3.4 or higher.

This entry was posted in **Database**, **Hacks**, **MongoDB**, **UNIX/Linux** and tagged **Database**, **Errors**, **Hacks**, **MongoDB**, **UNIX/Linux** by **Koen**. Bookmark the **permalink [http://koenaerts.ca/compile-and-install-mongodb-on-raspberry-pi/]** .

3 thoughts on “COMPILE AND INSTALL MONGODB ON RASPBERRY PI”

Ho

on **July 22, 2017 at 1:51 pm** said:

Thank you very much for this clear instruction. I can compile and run mongodb without any difficulty.

Daniel

on **August 27, 2017 at 2:48 am** said:

using gcc version 6.3.0 20170516 (Raspbian 6.3.0-18+rpi1)
I've got the following error while building:

```
In file included from src/mongo/platform/compiler.h:132:0,
from src/mongo/util/invariant.h:30,
from src/mongo/base/string_data.h:39,
from build/opt/mongo/base/error_codes.h:31,
from src/mongo/base/status.h:34,
from src/mongo/client/dbclient.cpp:36:
src/mongo/client/dbclientcursor.h: In member function 'void mongo::DBClientCursor::_assertIfNull() const':
src/mongo/util/assert_util.h:233:28: error: nonnull argument 'this' compared to NULL [-Werror=nonnull-compare]
```

Bharat

on **November 21, 2017 at 7:05 pm** said:

nonnull argument 'this' compared to NULL [-Werror=nonnull-compare]