MobilityDB distribution on cloud 1.0 User’s Manual

COLLABORATORS			
	TITLE : MobilityDB distribution on cloud 1.0 User’s Manual		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Sid Ahmed BOUZOUIDJA	June 22, 2021	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Abstract

MobilityDB is an extension to the PostgreSQL database system and its spatial extension PostGIS. It allows temporal and spatio-temporal objects to be stored in the database, that is, objects whose attribute values and/or location evolves in time. MobilityDB includes functions for analysis and processing of temporal and spatio-temporal objects and provides support for GiST and SP-GiST indexes. MobilityDB is open source and its code is available on Github. An adapter for the Python programming language is also available on Github.

MobilityDB is developed by the Computer & Decision Engineering Department of the Université Libre de Bruxelles (ULB) under the direction of Prof. Esteban Zimányi. ULB is an OGC Associate Member and member of the OGC Moving Feature Standard Working Group (MF-SWG).

This is the manual for MobilityDB v1.0.

Chapter 1

Introduction

MobilityDB is an extension of PostgreSQL and PostGIS that provides temporal types. Such types are data types that represent the evolution on time of values of some element type, called the base type of the temporal type. For instance, temporal integers may be used to represent the evolution on time of the number of employees of a department. In this case, the data type is “temporal integer” and “integer” is the base type. Similarly, a temporal float may be used to represent the evolution on time of the temperature of a room. As another example, a temporal point may be used to represent the evolution on time of the location of a car, as reported by GPS devices. Temporal types are useful because representing values that evolve in time is essential in many applications, for example in mobility applications. Furthermore, the operators on the base types (such as arithmetic operators and aggregation for integers and floats, spatial relationships and distance for geometries) can be intuitively generalized when the values evolve in time.

MobilityDB provides the following temporal types: `tbool`, `tint`, `tfloat`, `ttext`, `tgeompoint`, and `tgeogpoint`. These temporal types are based, respectively, on the `bool`, `int`, `float`, and `text` base types provided by PostgreSQL, and on the `geometry` and `geography` base types provided by PostGIS (restricted to 2D or 3D points).[1] Furthermore, MobilityDB uses four time types to represent extents of time: the `timestamptz` type provided by PostgreSQL and three new types which are `period`, `timestampset`, and `periodset`. In addition, two range types are defined in MobilityDB: `intrange` and `floatrange`.

1.1 Project Steering Committee

The MobilityDB Project Steering Committee (PSC) coordinates the general direction, release cycles, documentation, and outreach efforts for the MobilityDB project. In addition, the PSC provides general user support, accepts and approves patches from the general MobilityDB community and votes on miscellaneous issues involving MobilityDB such as developer commit access, new PSC members or significant API changes.

The current members in alphabetical order and their main responsibilities are given next:

- Mohamed Bakli: MobilityDB-docker, cloud and distributed versions, integration with Citus
- Krishna Chaitanya Bommakanti: MobilityDB SQLAlchemy, MEOS (Mobility Engine Open Source), pyMEOS
- Anita Graser: integration with Moving Pandas and the Python ecosystem, integration with QGIS
- Darafei Praliaskouski: integration with PostGIS
- Mahmoud Sakr: co-founder of the MobilityDB project, MobilityDB workshop, integration with pgRouting
- Esteban Zimányi (chair): co-founder of the MobilityDB project, overall project coordination, main contributor of the backend code, BerlinMOD benchmark, MobilityDB-python

[1] Although 4D temporal points can be represented, the M dimension is currently not taken into account.

1.2 Other Code Contributors

- Arthur Lesuisse
- Xinyiang Li
- Maxime Schoemans

1.3 Corporate Sponsors

These are corporate entities (in alphabetical order) that have contributed developer time or direct monetary funding to the MobilityDB project.

- Adonmo, India
- Innoviris, Belgium
- Université libre de Bruxelles, Belgium

1.4 Licensing

The following licenses can be found in MobilityDB:

Resource	**Licence**
MobilityDB code	PostgreSQL Licence
MobilityDB documentation	Creative Commons Attribution-Share Alike 3.0 License

1.5 Installation

1.5.1 Short Version

Extracting the tar ball

```
tar xvfz MobilityDB-1.0.tar.gz
cd MobilityDB-1.0
```

To compile assuming you have all the dependencies in your search path

```
mkdir build
cd build
cmake  ..
make
sudo make install
```

Once MobilityDB is installed, it needs to be enabled in each individual database you want to use it in.

```
createdb mobility
psql mobility -c 'CREATE EXTENSION PostGIS'
psql mobility -c 'CREATE EXTENSION MobilityDB'
```

1.5.2 Get the Sources

The MobilityDB latest release can be found in `https://github.com/MobilityDB/MobilityDB/releases/latest`

wget

To download this release:

```
wget -O mobilitydb-1.0.tar.gz https://github.com/MobilityDB/MobilityDB/archive/v1.0.tar.gz
```

Go to Section 1.5.1 to the extract and compile instructions.

git

To download the repository

```
git clone https://github.com/MobilityDB/MobilityDB.git
cd MobilityDB
git checkout v1.0
```

Go to Section 1.5.1 to the compile instructions (there is no tar ball).

1.5.3 Enabling the Database

MobilityDB is an extension that depends on PostGIS. Enabling PostGIS before enabling MobilityDB in the database can be done as follows

```
CREATE EXTENSION postgis;
CREATE EXTENSION mobilitydb;
```

Alternatively, this can be done in a single command by using `CASCADE`, which installs the required PostGIS extension before installing the MobilityDB extension

```
CREATE EXTENSION mobilitydb CASCADE;
```

1.5.4 Dependencies

Compilation Dependencies

To be able to compile MobilityDB, make sure that the following dependencies are met:

- GNU C compiler (`gcc`). Some other ANSI C compilers can be used but may cause problems compiling some dependencies such as PostGIS.
- GNU Make (`gmake` or `make`) version 3.1 or higher. For many systems, GNU make is the default version of make. Check the version by invoking `make -v`.
- PostgreSQL version 10 or higher. PostgreSQL is available from http://www.postgresql.org. Notice that for using SP-GiST indexes for MobilityDB you need at least PostgreSQL version 11.
- PostGIS version 2.5. PostGIS is available from https://postgis.net/. PostGIS version 3.0 or higher is currently **not supported**, this is planned for future releases of MobilityDB.
- GNU Scientific Library (GSL). GSL is available from https://www.gnu.org/software/gsl/. GSL is used for the random number generators.

Please notice that PostGIS has its own dependencies such as Proj4, GEOS, LibXML2, or JSON-C, and these libraries are also used in MobilityDB. For a full PostgreSQL/PostGIS support matrix and PostGIS/GEOS support matrix refer to http://trac.osgeo.org/postgis/wiki/UsersWikiPostgreSQLPostGIS.

Optional Dependencies

For user's documentation

- DocBook (`xsltproc`) is required for building the documentation. Docbook is available from http://www.docbook.org/.
- DBLatex (`dblatex`) is required for building the documentation in PDF format. DBLatex is available from http://dblatex.sourceforge.r

Example: Installing dependencies on Linux

Database dependencies

```
sudo apt-get install postgresql-12 postgresql-server-dev-12 postgresql-12-postgis
```

Build dependencies

```
sudo apt-get install cmake gcc libgsl-dev
```

1.5.5 Configuring

MobilityDB uses the cmake system to do the configuration. The build directory is different from the source directory.

To create the build directory

```
mkdir build
```

To see the variables that can be configured

```
cd build
cmake -L ..
```

1.5.6 Build and Install

Please notice that the current version of MobilityDB has only been tested on Linux systems. It may work on other UNIX-like systems, but remain untested. Support for Windows is planned. We are looking for volunteers to help us to test MobilityDB on multiple platforms.

The following instructions start from `path/to/MobilityDB` on a Linux system

```
mkdir build
cd build
cmake ..
make
sudo make install
```

When the configuration changes

```
rm -rf build
```

and start the build process as mentioned above.

1.5.7 Testing

MobilityDB uses ctest, the CMake test driver program, for testing. This program will run the tests and report results.

To run all the tests

```
ctest
```

To run a given test file

```
ctest -R '21_tbox'
```

To run a set of given test files you can use wildcards

```
ctest -R '22_*'
```

1.6 Support

MobilityDB community support is available through the MobilityDB github page, documentation, tutorials, mailing lists and others.

1.6.1 Reporting Problems

Bugs are reported and managed in an issue tracker. Please follow these steps:

1. Search the tickets to see if your problem has already been reported. If so, add any extra context you might have found, or at least indicate that you too are having the problem. This will help us prioritize common issues.
2. If your problem is unreported, create a new issue for it.
3. In your report include explicit instructions to replicate your issue. The best tickets include the exact SQL necessary to replicate a problem. Please also, note the operating system and versions of MobilityDB, PostGIS, and PostgreSQL.
4. It is recommended to use the following wrapper on the problem to pin point the step that is causing the problem.

```
SET client_min_messages TO debug;
<your code>
SET client_min_messages TO notice;
```

1.6.2 Mailing Lists

There are two mailing lists for MobilityDB hosted on OSGeo mailing list server:

- User mailing list: http://lists.osgeo.org/mailman/listinfo/mobilitydb-users
- Developer mailing list: http://lists.osgeo.org/mailman/listinfo/mobilitydb-dev

For general questions and topics about how to use MobilityDB, please write to the user mailing list.