Help Topic: Launching the MOOSDB

Spring 2020

Michael Benjamin, mikerb@mit.edu
Department of Mechanical Engineering
MIT, Cambridge MA 02139

Launching the MOOSDB

Here we describe how to launch the MOOSDB from the perspective of the first-time user. The MOOSDB application is a server that runs on the robot or unmanned vehicle computer, or simply on your laptop during simulation. It may be launched from the command line, assuming it is in your path. Two minimalist methods are described here, starting with the most bare-bones.

A Bare-Bones Launching of the MOOSDB

In a bare-bones manner, the MOOSDB may be launched from the command line without any arguments. Normally the MOOSDB needs to know at least two pieces of configuration information, (a) the machine IP address on which to run, and (b) the port number on which to serve clients.

(Recall an IP address is comprised of four numbers in the range of [0,255] separated by a period. IP addresses are used for finding a particular machine on the Internet. For example, the oceanai class server resides at 18.18.38.22. The term "localhost" acts as a stand-in IP address when refering to one's own machine. The port number refers to a channel of sorts on a computer. It's fair to think of the IP address like the street address of an apartment building, and the port number as the unit number within the building where a particular resident resides.)

When launching the MOOSDB with no command line arguments, it will default to running on the localhost and port 9000:

```
$ MOOSDB
------------------- MOOSDB V10 -------------------
Hosting  community                "#1"
Name look up is                   off
Asynchronous support is           on
Connect to this server on port    9000
--------------------------------------------------
network performance data published on localhost:9020
listen with "nc -u -lk 9020"
```

At this point the MOOSDB is running on the local machine, serving clients on port 9000. A few variables are already being published, by the MOOSDB itself. You can open a scope with uXMS in another terminal window:

```
$ uXMS DB_CLIENTS DB_TIME DB_UPTIME --serverhost=localhost --serverport=9000

============================================================
uXMS_581                                            0/0(154)
============================================================
VarName     (S)ource   (T)ime  (C)  VarValue (SCOPING:EVENTS)
----------  ---------  ------  ---  -------------------------
DB_CLIENTS  MOOSDB_#1  38.14        "uXMS_581,"
DB_TIME     MOOSDB_#1  38.14        1387380731.414707
DB_UPTIME   MOOSDB_#1  38.14        39.00859
```

The Source and Time columns may be expanded by hitting the "s" and "t" keys respectively after it launches, or you can add --show=source,time as a command line argument to uXMS, to launch with these two columns expanded.

Alternatively you can scope with the umm tool:

```
$ umm --spy
```

The umm tool is part of the MOOS project at Oxford, while uXMS tool is part of the MOOS-IvP project at MIT. Overlap between toolset functions can often be found.

A More Civilized Launching of the MOOSDB

Virtually all MOOS applications are launched with a "mission configuration" file, a.k.a. a "dot-moos" file. The below mission file, moosdb_alpha.moos, provides the minimal configuration parameters the MOOSDB likes to see upon starting.

```
// (wget http://oceanai.mit.edu/2.680/examples/moosdb_alpha.moos)
ServerHost = localhost
ServerPort = 9000
Community  = alpha
```

Passing the moosdb_alpha.moos file as a command line argument produces the below output.

```
$ MOOSDB moosdb_alpha.moos
------------------- MOOSDB V10 --------------------
  Hosting  community               "alpha"
  Name look up is                  off
  Asynchronous support is          on
  Connect to this server on port   9000
---------------------------------------------------
network performance data published on localhost:9020
listen with "nc -u -lk 9020"
```

Try this. Either copy and paste the above .moos file, or use wget as shown on the first line above in the example file. If you haven't installed wget or similar (e.g., curl), you should pause here and do that.

```
$ sudo apt install wget             (On Linux)
$ brew install wget                 (On MacOS using Homebrew)
$ sudo port install wget            (On MacOS using MacPorts)
```