Linux H2OLAB installation

Nadir SOUALEM – INRIA

September 9, 2010

1 Requirements

- g++ 4.3 or higher even if code is working for lower version (3.4.6 seems to work !!!), 4.4.x offers good optimization
- **cmake** version 2.4.8 or higher
- Subversion client **svn**
- Mpi implementation (Intel, Mpich2, OpenMpi, etc ..)

2 Getting H2OLAB sources

2.1 How to define environment variables

First of all, find the shell you use:
`echo $SHELL`
Each shell has its own configuration file.

`$SHELL`	bash,sh	ksh	zsh	csh,tcsh
File Configuration	/.bashrc	/.kshrc	/.zshrc	/.cshrc

To define environment variables, you must put them in the corresponding file configuration of your shell. Suppose you want to define the environment variable `MY_DIR` pointing to the absolute path */absolute/path/to/my/directory*.
For **csh**, **tcsh**, edit the corresponding file configuration and add at the end :

```
setenv MY_DIR /absolute/path/to/my/directory
```

For **bash**, **ksh**, **sh**, **zsh**, edit the corresponding file configuration and add at the end :

```
export MY_DIR="/absolute/path/to/my/directory"
```

In the following documentation, i will suppose that you have the **bash**, **ksh**, **sh** or **zsh** shell. For **csh** and **tcsh**, you can easily adapt by using the **setenv** command to export and the corresponding file configuration of your shell.

2.2 Create Directory structure

Create two folders *lib_ext*, *H2OLAB* and place new folder *svn* under *H2OLAB*

```
mkdir -p lib_ext H2OLAB/svn
```

2.3 Create H2OLAB environment variables

Define environment variables `$HYDROLAB_ROOT` and `LIB_EXT` pointing to their absolute path. Edit the corresponding file configuration and add at the end :

```
export HYDROLAB_ROOT="/your/absolute/path/to/H2OLAB"
export LIB_EXT="/your/absolute/path/to/lib_ext"
```

2.4 Load H2OLAB environment variables

Reload your session according your shell, for example if your shell is `bash`

```
source ~/.bashrc
```

2.5 Checkout code sources from Inria Gforge repository

Copy all sources of H2OLAB, from the GForge INRIA server to the folder *svn* and all sources of External Libraries to the folder *lib_ext*.
Note : do not forget the dots at the end.

```
cd $HYDROLAB_ROOT/svn
svn checkout svn+ssh://login@scm.gforge.inria.fr/svn/hydrolab/trunk .
cd $LIB_EXT
svn checkout svn+ssh://login@scm.gforge.inria.fr/svn/h2olibext/unix .
```

3 Configuration of your development environment

External libraries require **mpicc** and **mpicxx** commands dedicated to MPI execution.

3.1 Mpi is already installed on your environment

If MPI is already installed, just define environment variable `$MPI_ROOT` pointing to the absolute path of your MPI installation directory and add the bin subdirectory of the installation directory to your path. `$MPI_ROOT` must contain the include directory of Mpi.
Edit the corresponding file configuration and add at the end :

```
export MPI_ROOT="$LIB_EXT/mpi"
export PATH="$MPI_ROOT/bin:$PATH"
```

Reload your session according your shell, for example if your shell is bash

```
source ~/.bashrc
```

Note : Sometimes it happens that */bin* is not on `$MPI_ROOT`, then you must define the `PATH`, according your installation. Furthermore, in few clusters there are several implementations of MPI, by defining your `$PATH` here, you specify the prority use of the implementation you need.

3.2 Mpich2 installation

If MPI is already installed, i will explain here how to install mpich2.

3.2.1 Installation

Unpack the tar file in the directory `$LIB_EXT` and create a new folder *mpich2-install*, wherever you want

```
cd $LIB_EXT
tar -xvzf mpich2-1.0.8.tar.gz
mkdir mpich2-install
```

Change directory to mpich2-1.0.8 and configure MPICH2 with the absolute path of mpich2-install like follows

```
cd $LIB_EXT/mpich2-1.0.8
./configure --prefix=/your/absolute/path/to/mpich2-install
```

Build and install MPICH2

```
make && make install
```

3.2.2 Environment variable $MPI_ROOT

Define environment variable `$MPI_ROOT` pointing to the absolute path of your MPI installation directory and add the bin subdirectory of the installation directory to your path.
Edit the corresponding file configuration and add at the end :

```
export MPI_ROOT="$LIB_EXT/mpich2-install"
export PATH="$MPI_ROOT/bin:$PATH"
```

Reload your session according your shell, for example if your shell is bash

```
source ~/.bashrc
```

Check that everything is in order at this point by doing

```
which mpiexec
```

3.2.3 Launch the process manager mpd

First create a file *.mpdconf* on your `$HOME` and make it readable and writable only by you:

```
cd $HOME
touch .mpd.conf
chmod 600 .mpd.conf
```

Edit *.mpd.conf* and add the following line:

```
MPD_SECRETWORD=define_your_password_of_mpd
```

Note : It should NOT be your normal Unix password.
Launch **mpd** process:

```
mpd&
```

3.3 Installation of External Libraries

Once MPI is installed, we'll launch the batch script file *install*, located in *$LIB_EXT*

```
cd $LIB_EXT
sh install
```

External Libraries are installed.

3.4 Installation of H2OLAB

Launch the batch file script *$HYDROLAB_ROOT/svn/install/linux/install*. It will create an appropriate folders structure under `$HYDROLAB_ROOT` and create your own parameters files (in *$HYDROLAB_ROOT/runs*).

```
cd $HYDROLAB_ROOT/svn/install/linux
chmod +x install
./install
```

Do not forget to add in your login shell configuration .bashrc (for bash and sh) or .cshrc (for csh and tcsh)

```
source $HYDROLAB_ROOT/svn/install/linux/env_hydrolab
```

to have definitively the necessary environment variables and reload your session according your shell, for example if your shell is bash

```
source ~/.bashrc
```

H2OLAB is now installed

4 Compilation of H2OLAB

H2OLAB uses **cmake** to build automatic Makefiles. The default behaviour is the cluster one, if you launch **cmake** without options, the built project will get the following properties:

- Release mode
- without OPENGL

4.1 Compilation without IDE – native mode

4.1.1 Out-Source Building

This is the prefered mode. To use Release out-of-source build just create a directory **outside** of *svn* directory. For example to build PARADIS:

```
$ mkdir build
$ cd build
$ cmake $HYDROLAB_ROOT/svn
$ make PARADIS
```

You can also build libraries, or other executables:

```
$ make Utilitary_Basis
$ make RUNS_GENERATION
```

To specify Debug mode configuration, just replace **cmake** command-line by

```
cmake -DCMAKE_BUILD_TYPE=Debug $HYDROLAB_ROOT/svn
```

4.1.2 In-Source building

To use in-source build just go to `$HYDROLAB_ROOT/svn`:

```
$ cd $HYDROLAB_ROOT/svn
$ cmake .
$ make PARADIS
```

To specify Debug mode configuration, just replace **cmake** command-line by

```
cmake -DCMAKE_BUILD_TYPE=Debug .
```

4.2 Compilation with CodeBlocks IDE

4.2.1 Out-Source Building

To use Release out-of-source build just create a directory where it will get built:

```
$ mkdir build
$ cd build
$ cmake -G "CodeBlocks - Unix Makefiles" $HYDROLAB_ROOT/svn
$ codeblocks H2OLAB.cbp &
```

and select the buid target you want to compile.
To specify Debug mode configuration, just replace **cmake** command-line by

```
cmake -G "CodeBlocks - Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug
$HYDROLAB_ROOT/svn
```

4.2.2 In-Source building

To use in-source build just go to `$HYDROLAB_ROOT/svn`:

```
$ cd $HYDROLAB_ROOT/svn
$ cmake -G "CodeBlocks - Unix Makefiles"  .
$ codeblocks H2OLAB.cbp &
```

and select the buid target you want to compile.
To specify Debug mode configuration, just replace **cmake** command-line by

```
cmake -G "CodeBlocks - Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug .
```

4.3 Compilation with Eclipse and Kdevelop IDE

To specify Eclipse or Kdevelop IDE , just replace in the previous section option

```
-G "CodeBlocks - Unix Makefiles"
```

by

```
-G "Eclipse - Unix Makefiles"
```

or

```
-G "Kdevelop - Unix Makefiles"
```

Note : It is recommended to use CodeBlocks IDE, Eclipse and Kdevelop consume too much memory.

4.4 OPENGL visualization

If you want to use OPENGL mode just add:

```
-DWITH_OPENGL=1
```

to **cmake** command-line.

5 Execution of H2OLAB

Executables such *PARADIS* you have built are in *$HYDROLAB_ROOT/H2OLAB_INSTALLATION/$CONFIGURATION/bin* for example for Release Configuration:

```
$ cd $HYDROLAB_ROOT/H2OLAB_INSTALLATION/Release/bin
$ ls -l PARADIS
-rwx------    1 rsli003  sli        78142139 Nov 02 12:17 PARADIS
```

Libraries used for linking these executables are in *$HYDROLAB_ROOT/H2OLAB_INSTALLATION/$CONFIGURATION/lib*
To execute *PARADIS*

```
$ cd $HYDROLAB_ROOT/H2OLAB_INSTALLATION/Release/bin
$ ./PARADIS
```

with several nodes or processors, for example 4:

```
$ mpiexec -n 4 ./PARADIS
```

6 Update H2OLAB

6.1 Update your code

To update your code:

```
$ cd $HYDROLAB_ROOT/svn
$ svn update
```

Note : it happens sometimes that someone changes the project tree, so it is highly recommanded to clean your cache whe you update your sources.

6.1.1 Out-Source cleaning

Remove the cache from your *build* directory:

```
$ cd build
$ rm CMakeCache.txt
```

and reload **cmake** according your configuration (Debug, Release mode, etc...).

6.1.2 In-Source cleaning

Remove the cache from your *svn* directory:

```
$ cd $HYDROLAB_ROOT/svn
$ rm CMakeCache.txt
```

and reload **cmake** according your configuration (Debug, Release mode, etc...).