AUTHOR = Enrique Tobis
Last updated = Oct 7, 2010
MATLAB compiler

I believe I managed to get the MATLAB compiler working in orchestra, and I would like to share the details with you, in case you need it.

IMPORTANT: To be able to use MATLAB-compiled executables, you should add the line

```
export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/matlab/sys/os/glnxa64:/opt/
matlab/bin/glnxa64:/opt/matlab/sys/java/jre/glnxa64/jre/lib/amd64/
native_threads:/opt/matlab/sys/java/jre/glnxa64/jre/lib/amd64/jli:/
opt/matlab/sys/java/jre/glnxa64/jre/lib/amd64:/opt/matlab/runtime/
glnxa64
```

to the file .bashrc in your home (a.k.a. ~/.bashrc)

1) Create a .m file with your script. For example, testing.m, contaning

```
function testing(cmdlineargument)
  fprintf('%s',cmdlineargument);
```

I hope the name is clear enough, but just in case, cmdlineargument would be the first command-line argument you give to the executable. It can take several command-line arguments.

2) Compile it, by running

```
/opt/matlab/bin/mcc -R -nojvm -R -nodisplay -C -m testing.m
```

-R -nojvm and -R -nodisplay tell MATLAB not to use any graphics. I took me a while to get it working like this, so stick to text-based programs for your research. Mind you, it is still possible to plot and save the plots to a file. You just can't open windows.

-m tells the compiler to generate a "C executable." That's what you want.

-C is perhaps the most important option of all. You see, mcc will generate a "stand-alone executable", but there is a catch. There is a ctf file that bundles your code, the toolbox functions you use and a bunch of other stuff. To make life more "convenient", this ctf file gets embedded into the executable by default (unless you use -C). If the ctf file lives inside the executable, every time you run it, it gets extracted. And here is where the problems start: it gets extracted to a directory in your home with a sequential name (e.g. ctf.1, ctf.2, ctf.3, etc). If you are running multiple jobs on multiple hosts, all of which have your home mounted via NSF, then you can see how this is a recipe for disaster. So, we use the -C option.

Once the compilation is done (it takes a while...), you will get two important files: testing and testing.ctf You want to keep those in the same directory. Otherwise, you're in for trouble.

3) Suppose you have testing and testing.ctf in /home/et62/MATLABCompiler The first time you execute /home/et62/MATLABCompiler/testing you will get the output

```
Extracting CTF archive. This may take a few seconds, depending on the
size of your application. Please wait...
...CTF archive extraction complete.
```

plus whatever your program prints or does, and it will create the directory testing_mcr

in /home/et62/MATLABCompiler
IT IS VERY IMPORTANT that you always invoke testing using a FULL PATH. If you put it in $PATH, and just execute testing, it will look for testing.ctf in the directory in which you are in, and tell you it cannot find testing.ctf.

4) You're all set. The executable doesn't need to extract anything else, so it will just run. REMEMBER: don't put the executable in your PATH. Run it with a full path.

I hope this helps. I know it has some extra details, but they may help you find out what's going on if you didn't follow the instructions carefully ;)