Matlab

Detailed documentation about Matlab can be found on the vendor's website.

Licensing

Matlab is a licensed product that requires free licenses to execute, the licenses for Matlab are shared between all of FEUPs users. It's therefore possible for jobs on the cluster to fail because there are no free licenses. This can be avoided by using the Matlab compiler to compile your Matlab code into an executable. Running this executable doesn't require a license. Matlab has parallelization support through it's Parallel Computing Toolbox. However, this is a licensed option that FEUP does not currently have. Therefore, Matlab rarely makes use of the multiple cores available on the cluster nodes.

Example usage

Example matlab script (example.m):

```
x = 'Hello world!';
disp(x);
exit;
```

To run the matlab script:

```
$ module load matlab
$ matlab -nodesktop -nodisplay -r example
```

Do not add the “.m” suffix to your file. Matlab adds this for you, so if you include it Matlab with try to open the file *example.m.m***!**
The *-nodesktop* **and** *-nodisplay* **arguments are not required, but prevent some warnings from appearing.**

Matlab compiler

Assuming the example.m script from above is in your current directory, you can compile it to a script with the command:

```
$ module load matlab
$ mcc -m example
```

After the command finishes, it will have produced 4 files:

- example
- mccExcludedFiles.log
- readme.txt
- run_example.sh

To use the compiled program, execute the *run_example.sh* script passing it the path to Matlab:

```
$ ./run_example.sh /soft/MATLAB/R2017b/
-----------------------------------------
Setting up environment variables
---
LD_LIBRARY_PATH is
.:/soft/MATLAB/R2017b//runtime/glnxa64:/soft/MATLAB/R2017b//bin/glnxa64:/sof
t/MATLAB/R2017b//sys/os/glnxa64:/soft/MATLAB/R2017b///sys/java/jre/glnxa64/j
re/lib/amd64/native_threads:/soft/MATLAB/R2017b//sys/java/jre/glnxa64/jre/li
b/amd64/server:/soft/MATLAB/R2017b//sys/java/jre/glnxa64/jre/lib/amd64/clien
t:/soft/MATLAB/R2017b//sys/java/jre/glnxa64/jre/lib/amd64
Warning: No display specified.  You will not be able to display graphics on
the screen.
Hello world!
```

Coder

MATLAB has a toolbox called Coder which allows you to transform Matlab code into C. According to MATLAB it allows you to “accelerate computationally intensive portions of MATLAB code and verify the behaviour of the generated code”.

If you have access to the Coder toolbox but are not running on the same OS as the cluster, you can generate the C code and transfer it to the cluster to compile and run. The tool to do this is called mex and instructions can be found in the MATLAB documentation.

The following example shows how to use it. The file *yprime.c* is provided by MATLAB:

```
$ module load matlab
$ mkdir test
$ cd test
$ cp /soft/MATLAB/R2017b/extern/examples/mex/yprime.c ./
$ mex yprime.c

$ cat <<EOF > test.m
yprime(1,1:4)
exit
EOF
$ matlab -nodesktop -nodisplay -r test
< M A T L A B (R) >
Copyright 1984-2017 The MathWorks, Inc.
R2017b (9.3.0.713579) 64-bit (glnxa64)
September 14, 2017


To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.


ans =
```

```
    2.0000    8.9685    4.0000   -1.0947

$
```

From:
https://grid.fe.up.pt/dokuwiki/ - **GRID FEUP**

Permanent link:
https://grid.fe.up.pt/dokuwiki/doku.php?id=documentation:software:matlab

Last update: **2018/06/25 18:16**