

Matlab on multi-processor machines

The aim of this short guide is to get you started with using matlab in a multi-processor environment. The goal is to instruct matlab to use a `pool' of n processors rather than a single one to speed up calculations.



Before trying to parallelize any codes, please make sure your code works as expected on one processor.



matlab versions prior to 2007 do not have the features discussed here.

Parallel Computing Toolbox aka matlabpool

matlabpool can be used to parallelize your m script across multiple cores in a single machine.

Open matlab and initialize the parallel environment

```
matlab -nodesktop

>> matlabpool ( 'open', 'local', 8)
Starting matlabpool using the 'local' configuration ... connected to 8 labs.
>> p=findResource('scheduler', 'configuration', 'local');

p =

Local Scheduler Information
=====

                        Type : local
          ClusterOsType : unix
            ClusterSize : 8
          DataLocation :
/home/lenocil/.matlab/local_scheduler_data/R2010b
      HasSharedFilesystem : true

- Assigned Jobs

      Number Pending   : 0
      Number Queued    : 0
      Number Running   : 1
      Number Finished  : 0

- Local Specific Properties

      ClusterMatlabRoot : /software/matlab/matlab-R2010b
```

The `matlabpool` cmd instructs matlab that you intend to run your program in on multiple processors. In the example above you are using a parallel environment called `'local'` (this is usually the default. See below on how to change its properties or create a custom one.) requesting 8 processors. `'local'` allocates resources that can be listed using the cmd `findResource`.

At this point you can start programming in matlab as usual. To make sure you can fully utilize the resources that `matlabpool` allocates, do not forget to write your loops using `parfor`. Once finished remember to close `matlabpool`

```
matlabpool('open',4); % opens the default parallel environment, usually  
called 'local'  
parfor k = 1 : 100000  
    some code  
end  
matlabpool close % release the resources
```

Gotchas

- `'parfor'` executes loop in an order that suits matlab, that is not sequentially
- `parfor` loops cannot be nested

Useful readings

http://vtchl.uiuc.edu/sites/default/files/MATLAB_Report.pdf

From:

<https://helpdesk.strw.leidenuniv.nl/wiki/> - **Computer Documentation Wiki**

Permanent link:

https://helpdesk.strw.leidenuniv.nl/wiki/doku.php?id=matlab_on_cluster&rev=1464939921

Last update: **2016/06/03 07:45**

