MATLAB Compiler Runtime (MCR)

Name	APPS/MATH/MCR-R2013A
Description	Matlab Compile Runtime
Status	Production
Last update	2013-10-25

USAGE

Due to licensing issues Matlab can't be directly use in the FGI environment. However, many Matlab codes can be translated to C coded executables. This executable can be used in FGI when the Matlab Compiler Runtime (MCR) is available. When Matlab code is executed this way it does not require license and thus large amounts of C converted Matlab codes can be executed simultaneously.

MCR-R2013A Runtime Environment supports c-converted Matlab codes that have been created with the *mcc* command of Matlab version R2013A (other Matlab versions are not compatible with the Runtime Environment).This Matlab version is available in the taito.csc.fi server of CSC.

A simple Matlab code (*testjob.m*) is converted to C exeutable with command: (download the sample case from here):

```
mcc -mv testjob.m
```

The conversion command produces C executable (*testjob*) and a shell scrip (*run_testjob.sh*) that can be used to launch the executable. These components, as well as any input files that the Matlab code uses, needs to be defined as input files in the grid job description file. In this sample case we don't have any input files so the job description file (*matlab.xrsl*) could look like following:

```
&
(executable=matlabrun.sh)
(jobname=matlab)
(stdout=std_1.out)
(stderr=std_1.err)
(walltime=4h)
(memory=1052)
(runtimeenvironment="APPS/MATH/MCR-R2013A")
(inputfiles=
( "testjob" "testjob" )
( "run_testjob.sh" "run_testjob.sh" )
)
(outputfiles=
  ( "matlabresults.txt" "matlabresults.txt" )
)
```

 In the command script (*matlabrun.sh*) execution permissions must first be defined for the compiled command and execution script. After this the compiled Matlab code is launched using the execution script (*run_testjob.sh*). The location of the MCR library must be given as the first argument for execution script. When the MCR-R2013A Runtime Environment is used, this location is stored to environment variable: $MCRROOT. The possible arguments of the Matlab code are added to the execution command after this.

In cases where large amounts of Matlab jobs are submitted it is advices to run the actual matlab task in the local temporary directory ($TMPDIR) of the computing node. To do this, the execution script must have steps where the input data is coped to $TMPDIR and the process is moves to the $TMPDIR before the actual execution of the Matlab task. The results must also be copied back to the original job directory when the actual matlab task has finished.

In this example the command script could look like following. :

```
#!/bin/sh
echo "Hello Matlab!"

#copy data to temporary directory
jobdir=`pwd`
cp -r * $TMPDIR/
cd $TMPDIR

#set execution permissions for the matlab script and executable
chmod u+x run_testjob.sh
chmod u+x testjob

./run_testjob.sh $MCRROOT > matlabresults.txt
exitcode=$?
```

```
#Copy results from the temporary directory
#back to the home directory of the grid job
mv matlabresults.txt $jobdir/

echo "Bye Matlab!"
exit $exitcode
```

Commands to submit the sample job in linux and MacOSX machines

Set up grid environment and submit the job

```
arcproxy
arcsub matlab.xrsl
```

Follow the progress of the job and retrieve the result when the job is finished,

```
arcstat job_id
arcget job_id
```

In the commands above, job_id should be replaced with the job identifier given by the the *arcsub* command for example:

```
arcstat gsiftp://grid.triton.aalto.fi:2811/jobs/2864513391466861182312366
arcget gsiftp://grid.triton.aalto.fi:2811/jobs/2864513391466861182312366
```

The arcget command will store the results of the job to a new directory named according to the job ID number.

Interface definition

The runtime environment sets the following environment variables:

- MCRROOT is set so that it points tro the MCR libraray location

System administrator guide for installing the RE

Download and install the MCR installation kits from:

http://www.mathworks.se/products/compiler/mcr/index.html

Runtime environmen

Download runtime environment example for SLURM,

Modify the scripts as needed and save the main script in your ARC runtime directory as APPS/MATH/MRC-R2013A

As long as the interface requirements are satisfied, the implementation does not really matter. And some adaptation is needed anyway to accommodate differences in the cluster environment (batch queue systems, temporary directory location etc.)