

MATLAB – Execution Guidelines for running applications in aggregated environment using ScaleMP’s vSMP Foundation

Overview

Matlab is using Intel’s Math Kernel Library (Intel® MKL), which offers highly optimized extensively threaded math routines for scientific, engineering, and financial applications that require maximum performance.

Matlab 2007b and later versions support Multi-Threading. Matlab 2007b limits the maximum number of threads to 16, not the total number of cores on the system. Matlab 2009a limits the maximum number of threads to 32

By using MKL version 10 and above, it is possible to overcome the Matlab threading limit by instructing MKL directly on how many threads to use. This is done by setting the MKL_NUM_THREADS environment variable from within the Matlab script to the desired number of threads.

Running Matlab jobs with vSMP Foundation, enables running large memory jobs, taking advantage of the entire system memory.

Running Matlab

In the run-script for running Matlab set the number of threads to run with:

```
export NPROC=32
```

Assigning specific CPUs to threads is also highly recommended. Use KMP_AFFINITY to set the correct affinity:

```
export KMP_AFFINITY=verbose,compact
```

It will place first thread to CPU #0, next thread to CPU #1, and so on.

Also important is to tell MKL to honor the requested number of threads by instructing it not to choose how many threads to use dynamically:

```
export MKL_DYNAMIC=FALSE
```

In the Matlab script, set the number of threads as follows:

```
maxNumCompThreads (<number>);
```

Note that if the number is larger than the Matlab thread limit, the run will abort. To set the limit higher (with MKL >=10), add the following script code:

```
setenv('MKL_NUM_THREADS', getenv('NPROC'));
```

Important: in Matlab 2007b, the default number of threads is 1. To enable multi-threading, a call to maxNumCompThreads, with a value larger than one, must be performed before setting the environment variable.

Upgrading MKL in Matlab

Matlab 2007b uses MKL version 9, while Matlab 2009a used MKL version 10. Matlab provides a way to “upgrade” or use a different MKL or BLAS library. Under the Matlab installation folder the folder <MATLAB folder>/bin/glnxa64 contains the shared libraries (including MKL) and a text file named **blas.spec** which tells Matlab what BLAS library to use. The default file contains these two lines:

```
GenuineIntel Family * Model * mkl.so mklcompat.so # EM64T
```

```
AuthenticAMD Family * Model * acml.so acmlcompat.so # AMD64
```

To use a different version of MKL with MATLAB, comment the existing lines (with a '#' at the beginning) and add the following line:

```
GenuineIntel Family * Model * libmkl_core.so libmkl_intel_thread.so  
libmkl_intel_lp64.so libguide.so
```

There is also the need to perform a one time copy of the MKL libraries from the MKL location:

```
cp -p <MKL folder>/lib/em64t/libmkl*.so <MATLAB folder>/bin/glnxa64
```

Copying the libraries and setting blas.spec is enough to get Matlab 2007b to work with the upgraded MKL. However, Matlab 2009a requires also that some libraries be preloaded:

```
export LD_PRELOAD=<MATLAB-folder>/bin/glnxa64/libmkl_core.so:<MATLAB-  
folder>/bin/glnxa64/libguide.so:<MATLAB-  
folder>/bin/glnxa64/libmkl_intel_thread.so
```

With vSMP Foundation the following MKL versions are recommended:

- 10.0.2.018 for Harpertown based systems.
- 11.0.074 for Nehalem based systems.

Sample Run-Script

```
----- matmul.m -----
nproc=str2num(getenv('NPROC'));
maxNumCompThreads(32);          % Enable multi-threading

setenv('MKL_NUM_THREADS', getenv('NPROC'));

% Set parameters
numRuns = 1;                    % Number of runs to average over
dataSize = 20000;              % Data size to test
x=rand(dataSize,dataSize);    % Random square matrix

% Matrix multiplication
tic;
for i = 1:numRuns
    y=x * x;
end
toc
-----

----- runmatlab2007.sh -----
export NPROC=32

export MKL_DYNAMIC=FALSE

export KMP_AFFINITY=verbose,compact

matlab -nodisplay -nojvm -r matmul 2>&1 | tee output.txt
-----

----- runmatlab2009.sh -----
export NPROC=32

export MKL_DYNAMIC=FALSE

export KMP_AFFINITY=verbose,compact

MATLAB=/opt/matlab/R2009a

export
LD_PRELOAD=$MATLAB/bin/glnxa64/libmkl_core.so:$MATLAB/bin/glnxa64/libguide.so
:$MATLAB/bin/glnxa64/libmkl_intel_thread.so

matlab -nodisplay -nojvm -r matmul 2>&1 | tee output.txt
-----
```