OPENMP

Hands On

Installing and running C/C++/Fortran Programs on multicore machines

The GNU Project https://gcc.gnu.org/install/binaries.html

MacOS:

The Homebrew package manager : https://brew.sh/

MacPorts https://www.macports.org/

MS Windows

The Cygwin project https://sourceware.org/cygwin/

MinGW and mingw-w64 projects http://www.mingw.org/ http://mingw-w64.org/doku.php

Linux machines come pre installed with gnu binaries

```
gcc --version

gfortran --version
```

Environment Variables

LINUX /UNIX bash /Cygwin:
List all environment variables: **printenv**
Update environment variable : **export OMP_NUM_THREADS=5**

LINUX/UNIX csh
List all environment variables: **printenv**
Update environment variable : **setenv OMP_NUM_THREADS 5**

Windows/DOS
List all environment variables: **set**
Update environment variable : **set** (/A) **OMP_NUM_THREADS= 5**

OSX
List all environment variables: **env**
Update environment variable : **export OMP_NUM_THREADS= 5**

Compiling and running OPENMP Code

Locally

```
$g++ -fopenmp Program.cpp –o <output_name>
$gfortran –fopenmp Program.f95 –o <output_name>
$./<output_name>
```

On basic clusters at SERC (tyronne, delta)

```
$ssh <username>@delta-cluster.serc.iisc.ernet.in
$password:
```

Copy the code onto the home area
Create make files for compiling code (clearing binaries, compiling, linking and creating executable)

Run **make**
Create a job script to submit the job in batch mode
qsub jobscript
qstat
Check output

OpenMP environment variables

OMP_DYNAMIC : When set to TRUE, the number of threads available for executing parallel regions can be adjusted at run time to make the best use of system resources.

OMP_NESTED: When set to **TRUE**, nested parallelism is enabled. This means that the runtime environment might deploy extra threads to form the team of threads for the nested parallel region.

OMP_PROC_BIND : When TRUE, binds threads to processors.

OMP_MAX_ACTIVE_LEVELS: The maximum number of active nested parallel regions.

OMP_STACK_SIZE: Specifies the size of the stack for threads that are created by the OpenMP run time. (default in KB). If the compiler cannot deliver the stack size specified by the environment variable, or if **OMP_STACKSIZE** does not conform to the valid format, the compiler sets the environment variable to the default value.

OMP_WAIT_POLICY: Use **ACTIVE** if you want waiting threads to mostly be active. That is, the threads consume processor cycles while waiting. For example, waiting threads can spin while waiting. The **ACTIVE** wait policy is recommended for maximum performance on the dedicated machine. Use **PASSIVE** if you want waiting threads to mostly be passive. That is, the threads do not consume processor cycles while waiting. For example, waiting threads can sleep or yield the processor to other threads.

OpenMP environment variables

Display OPENMP environment upon execution of the program

$export OMP_DISPLAY_ENV=TRUE (bash)

$setenv OMP_DISPLAY_ENV TRUE (csh)

```
OPENMP DISPLAY ENVIRONMENT BEGIN
  _OPENMP = '201307'
  OMP_DYNAMIC = 'FALSE'
  OMP_NESTED = 'FALSE'
  OMP_NUM_THREADS = '32'
  OMP_SCHEDULE = 'DYNAMIC'
  OMP_PROC_BIND = 'FALSE'
  OMP_PLACES = ''
  OMP_STACKSIZE = '140729178218216'
  OMP_WAIT_POLICY = 'PASSIVE'
  OMP_THREAD_LIMIT = '4294967295'
  OMP_MAX_ACTIVE_LEVELS = '2147483647'
  OMP_CANCELLATION = 'FALSE'
  OMP_DEFAULT_DEVICE = '0'
OPENMP DISPLAY ENVIRONMENT END
```