OpenMP implementation for FORTRAN on HPC

Speaker: Dr **Liang Sun**

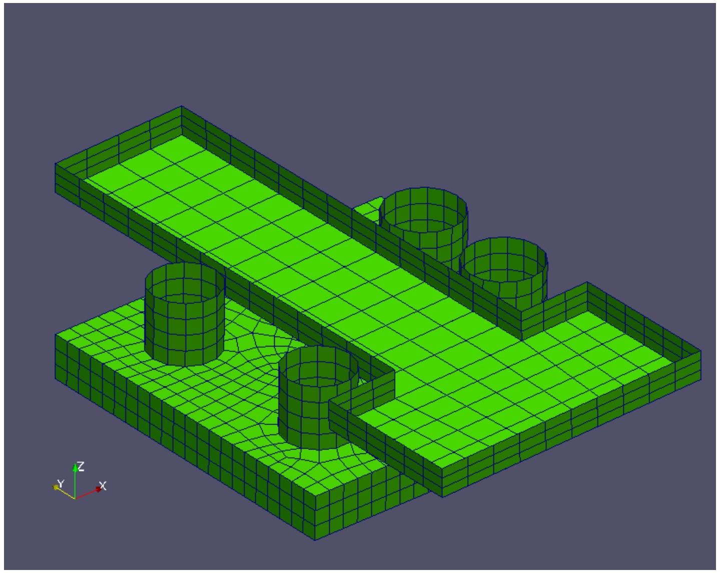
Email: *l.sun@bath.ac.uk*

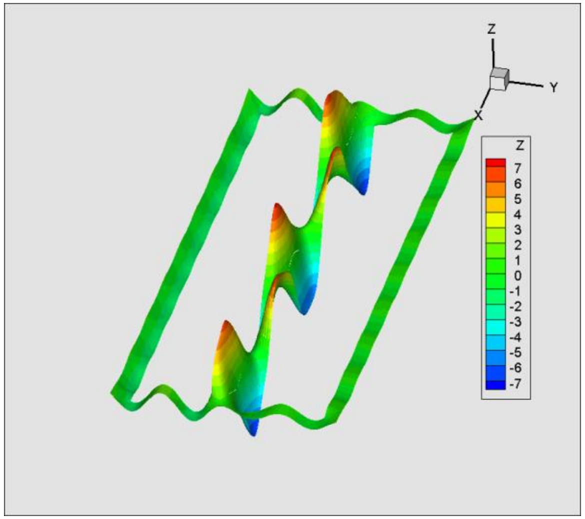
Co-author: Dr **Jun Zang**

Email: *j.zang@bath.ac.uk*

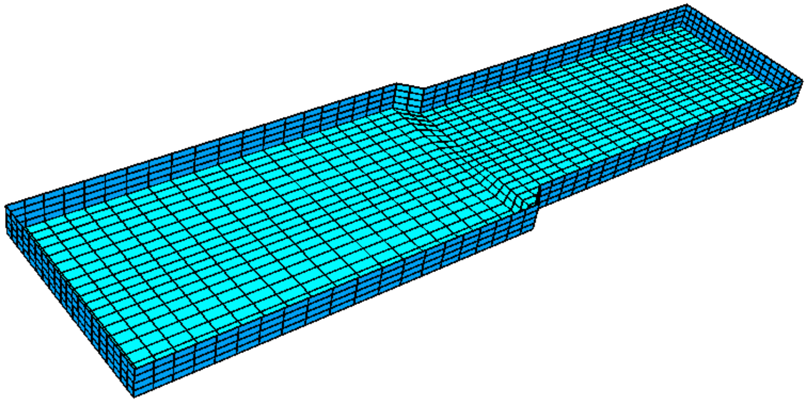
Department of Architecture and Civil Engineering
University of Bath

Background



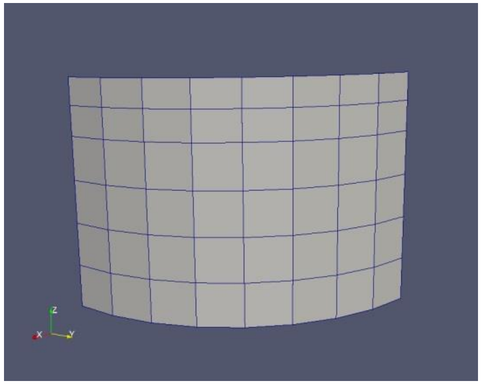


Numerical method

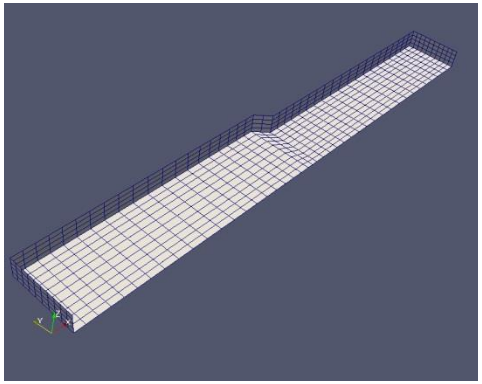


$$[A]\{x\}=\{b\}$$

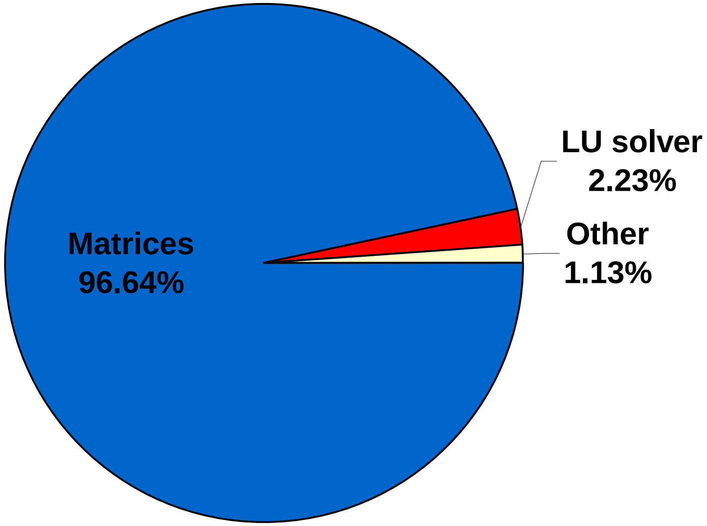
Performance of sequential executable

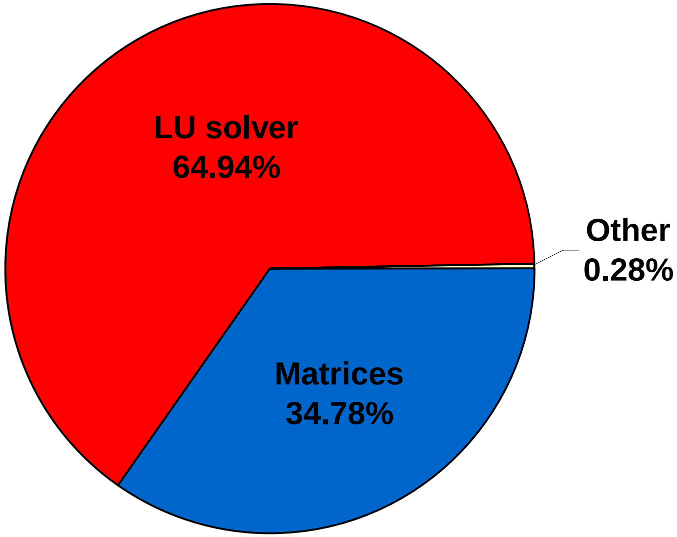


$[A]_{264\times264}$

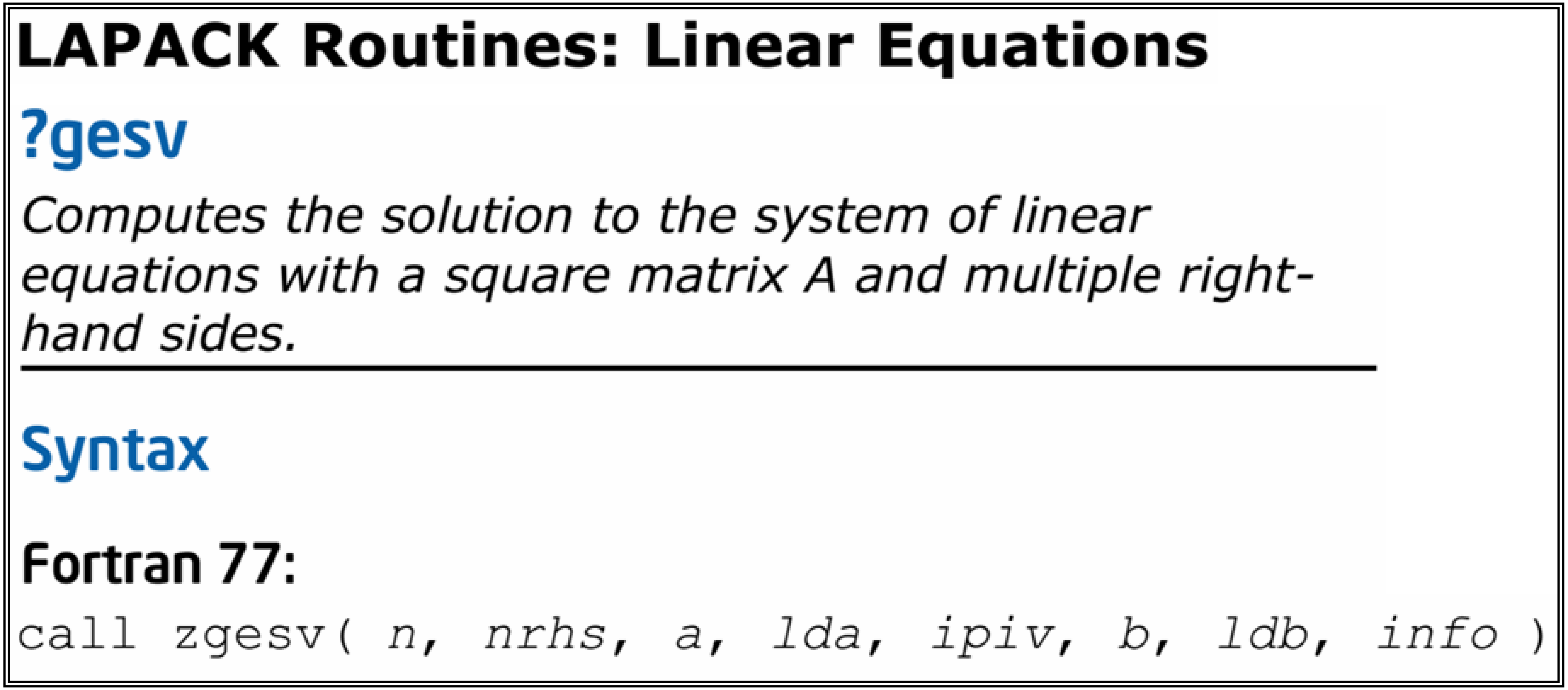


$[A]_{3952\times3952}$





Better LU solver in Intel MKL



LAPACK Routines: Linear Equations

?gesv

Computes the solution to the system of linear equations with a square matrix A and multiple right-hand sides.

Syntax

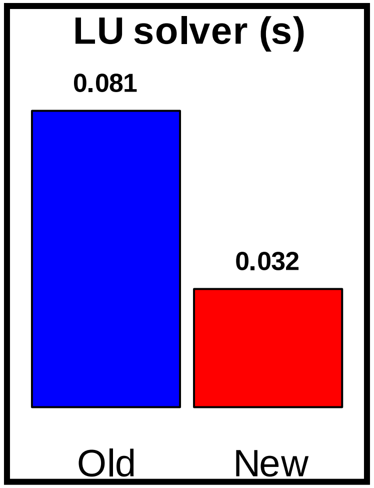
Fortran 77:

```
call zgesv( n, nrhs, a, lda, ipiv, b, ldb, info )
```

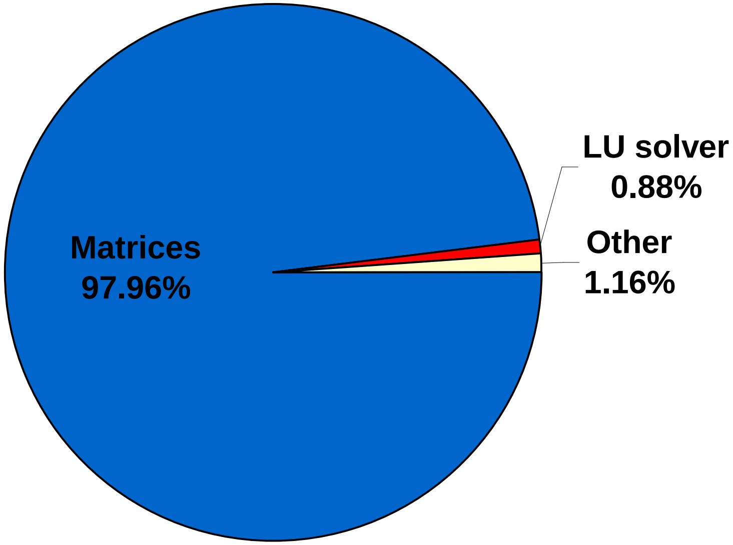
Reference:
Intel® Math Kernel Library Reference Manual
http://software.intel.com/en-us/articles/intel-math-kernel-library-documentation

Performance of sequential executable with MKL

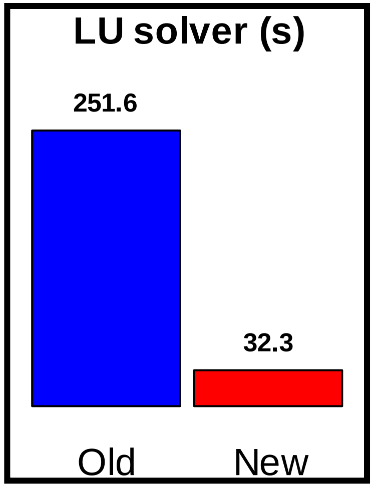
$$[A]_{264\times264}$$



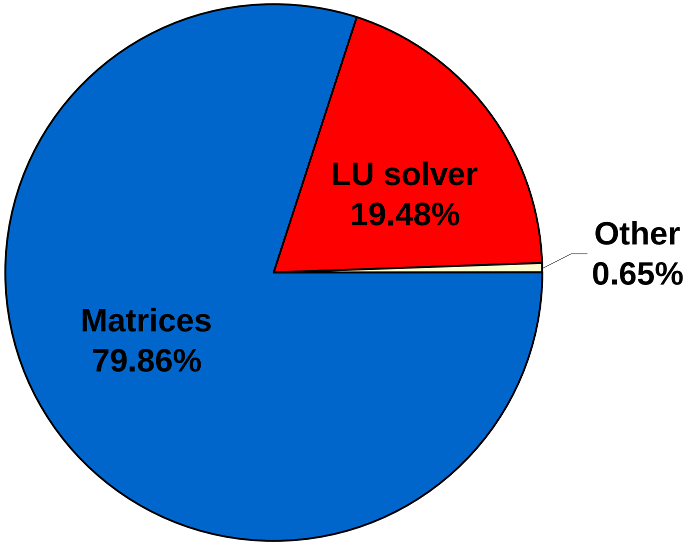
save **60%**



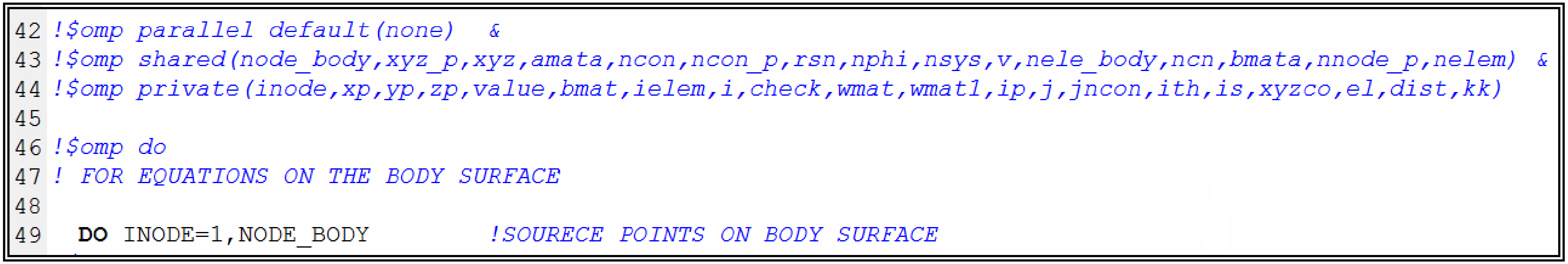
$$[A]_{3952\times3952}$$



save **87%**



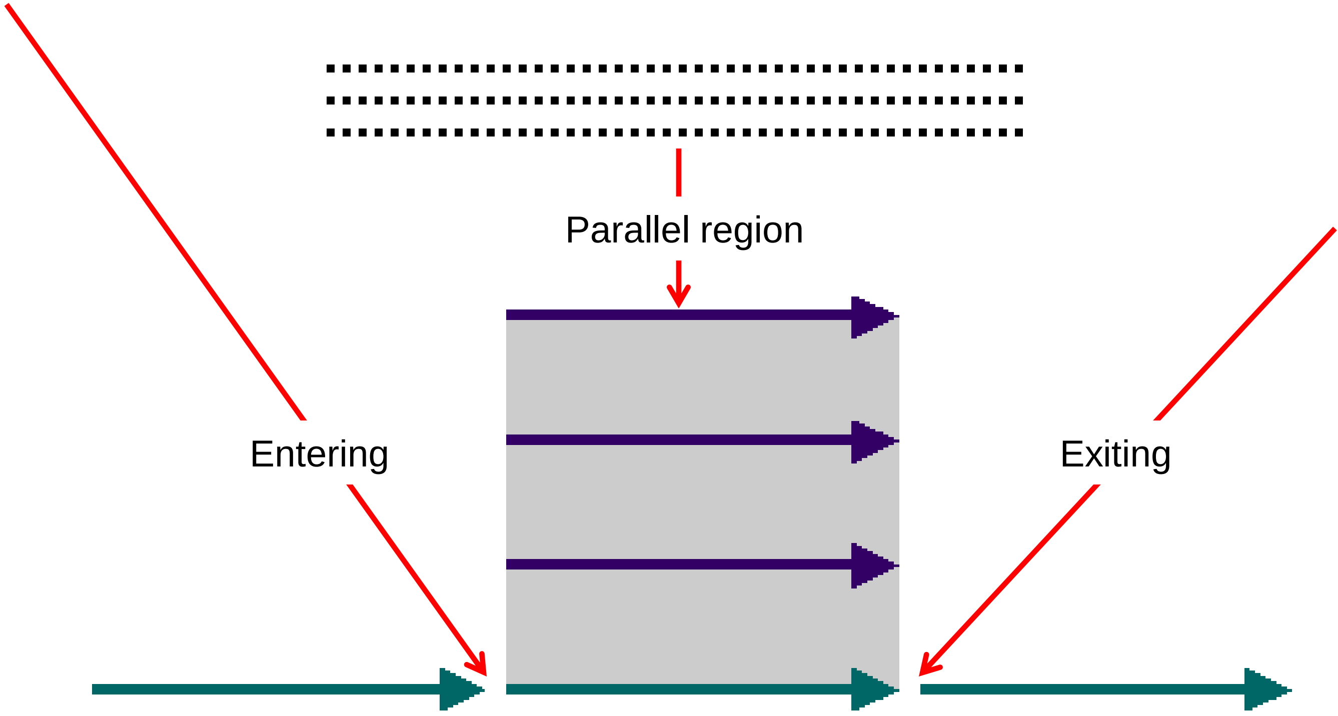
Implementation of OpenMP

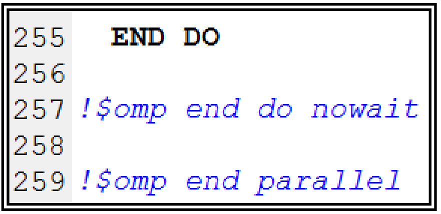


```
!$omp parallel default(none)  &
!$omp shared(node_body,xyz_p,xyz,amata,ncon,ncon_p,rsn,nphi,nsys,v,nele_body,ncn,bmata,nnode_p,nelem) &
!$omp private(inode,xp,yp,zp,value,bmat,ielem,i,check,wmat,wmat1,ip,j,jncon,ith,is,xyzco,el,dist,kk)

!$omp do
! FOR EQUATIONS ON THE BODY SURFACE

  DO INODE=1,NODE_BODY          !SOURECE POINTS ON BODY SURFACE
```





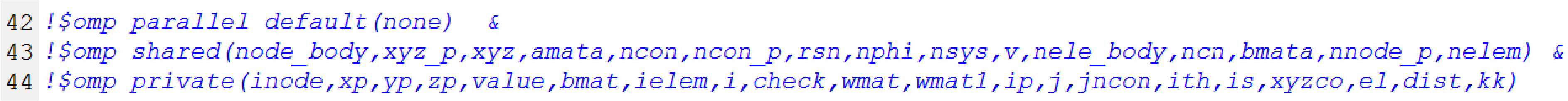
```
   END DO

!$omp end do nowait

!$omp end parallel
```

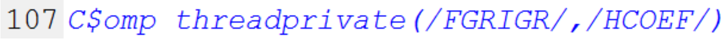
Data race

➢ Data-Sharing Attributes



```
!$omp parallel default(none)   &
!$omp shared(node_body,xyz_p,xyz,amata,ncon,ncon_p,rsn,nphi,nsys,v,nele_body,ncn,bmata,nnode_p,nelem)  &
!$omp private(inode,xp,yp,zp,value,bmat,ielem,i,check,wmat,wmat1,ip,j,jncon,ith,is,xyzco,el,dist,kk)
```

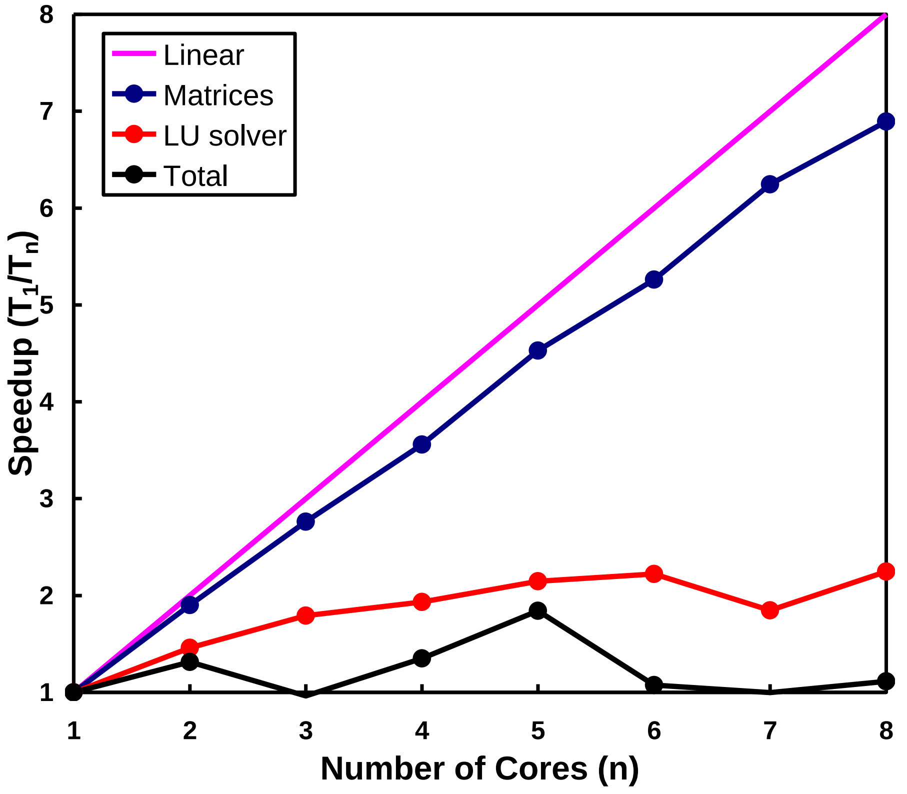
➢ Threadprivate Directive



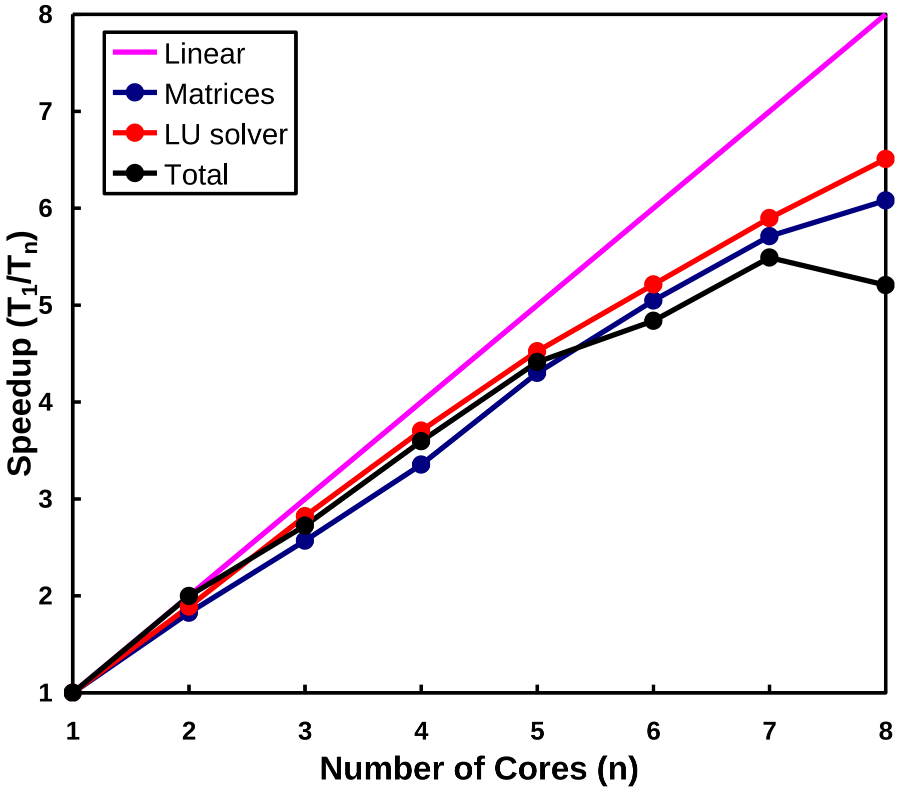
```
C$omp  threadprivate(/FGRIGR/,/HCOEF/)
```

Speedup of parallel executable

$[A]_{264\times 264}$



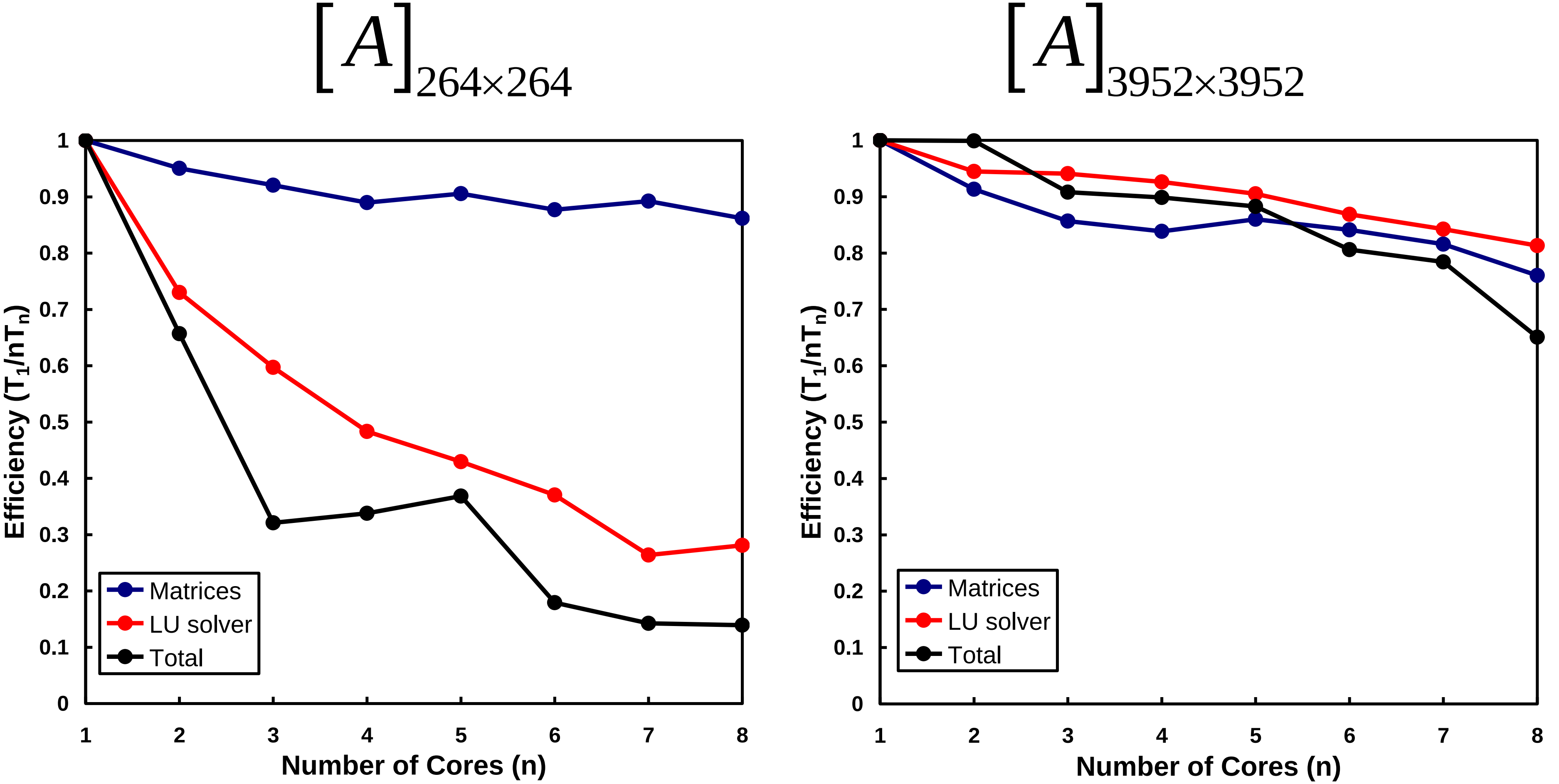
$[A]_{3952\times 3952}$



$\mathbf{T_1}$ is the execution time of the sequential algorithm

$\mathbf{T_n}$ is the execution time of the parallel algorithm with *n* cores

Efficiency of parallel executable



$$[A]_{264\times264}$$

$$[A]_{3952\times3952}$$

T_1 is the execution time of the sequential algorithm

T_n is the execution time of the parallel algorithm with *n* cores

Concluding remarks

- Optimized LU solver in Intel MKL improves performance significantly

- OpenMP has been implemented successfully in current FORTRAN codes and all data race problems have been solved

- Running multithreaded executable for small problems is not economical considering total computational time. In large problems, much time can be saved by using parallel algorithm

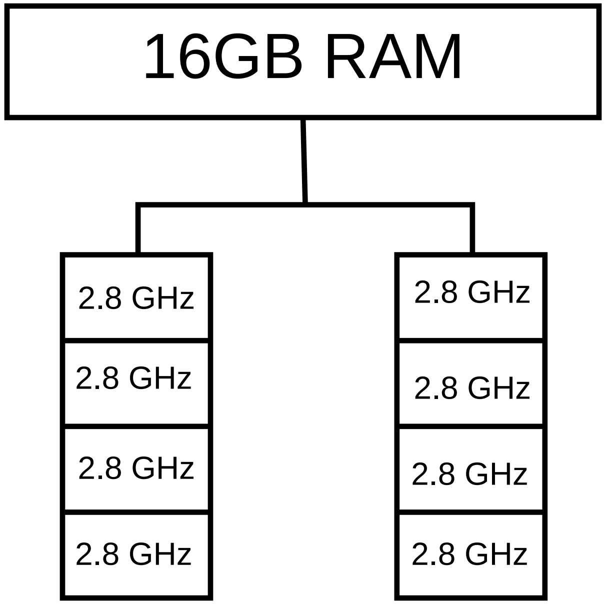
Acknowledgement

These computations were performed on the University of Bath's High Performance Computing Facility. Provision of services by BUCS HPC Support Team is gratefully acknowledged.

Thank You !

Additional Information

Hardware and software on HPC



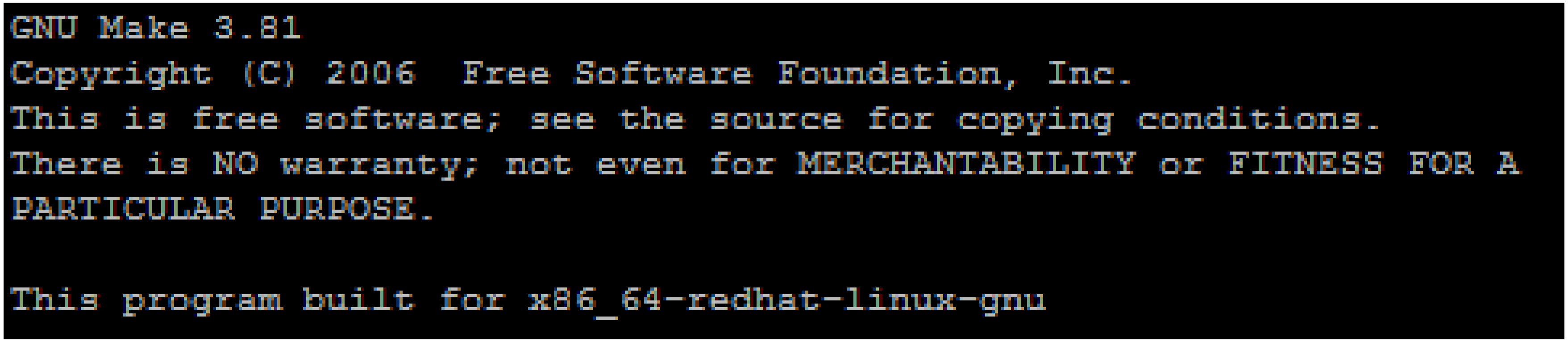
HPC Node

Intel FORTRAN Compiler
Module: *icomp/11.1.075*

Intel Math Kernel Library (MKL)
Module: *imkl/10.2.7.041*

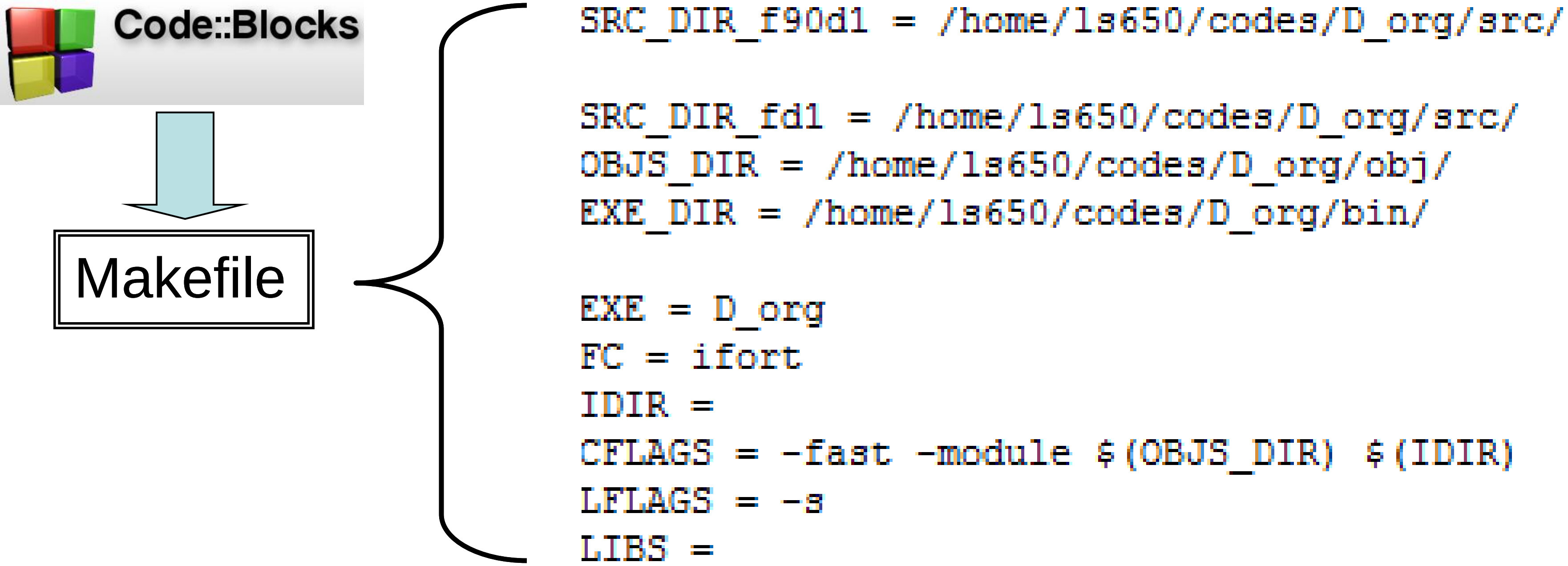
Compiler and Library

Generation of sequential executable



```
GNU Make 3.81
Copyright (C) 2006  Free Software Foundation, Inc.
This is free software; see the source for copying conditions.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

This program built for x86_64-redhat-linux-gnu
```

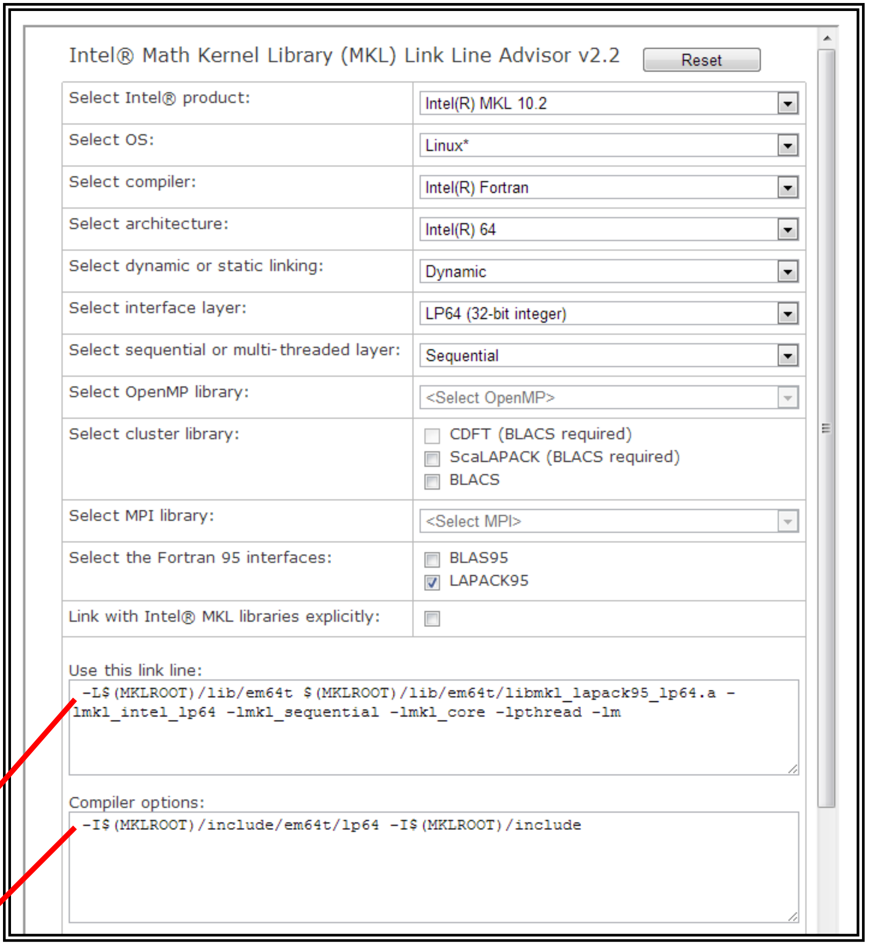


```
SRC_DIR_f90d1 = /home/ls650/codes/D_org/src/

SRC_DIR_fd1 = /home/ls650/codes/D_org/src/
OBJS_DIR = /home/ls650/codes/D_org/obj/
EXE_DIR = /home/ls650/codes/D_org/bin/

EXE = D_org
FC = ifort
IDIR =
CFLAGS = -fast -module $(OBJS_DIR) $(IDIR)
LFLAGS = -s
LIBS =
```

Generation of sequential executable with MKL



Intel® Math Kernel Library (MKL) Link Line Advisor v2.2

Select Intel® product:	Intel(R) MKL 10.2
Select OS:	Linux*
Select compiler:	Intel(R) Fortran
Select architecture:	Intel(R) 64
Select dynamic or static linking:	Dynamic
Select interface layer:	LP64 (32-bit integer)
Select sequential or multi-threaded layer:	Sequential
Select OpenMP library:	<Select OpenMP>
Select cluster library:	□ CDFT (BLACS required) □ ScaLAPACK (BLACS required) □ BLACS
Select MPI library:	<Select MPI>
Select the Fortran 95 interfaces:	□ BLAS95 ☑ LAPACK95
Link with Intel® MKL libraries explicitly:	□

Use this link line:

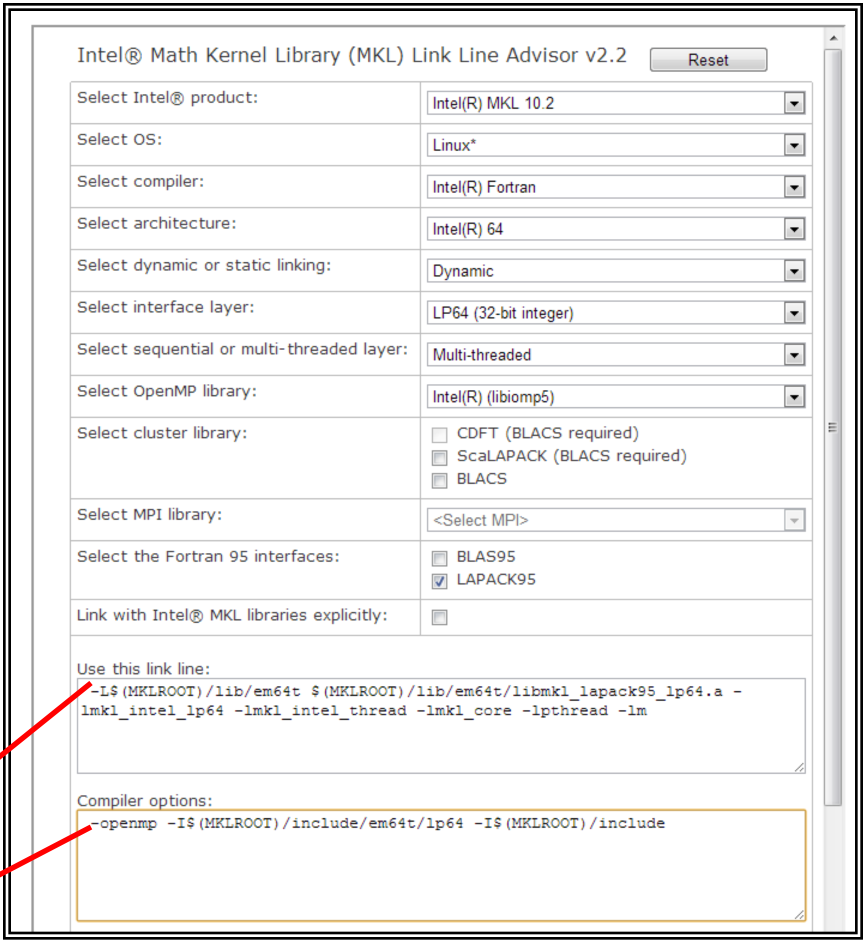
```
-L$(MKLROOT)/lib/em64t $(MKLROOT)/lib/em64t/libmkl_lapack95_lp64.a -
lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm
```

Compiler options:

```
-I$(MKLROOT)/include/em64t/lp64 -I$(MKLROOT)/include
```

```
CFLAGS = -fast [redacted] -module $(OBJS_DIR) $(IDIR)
LFLAGS = -s
LIBS = -L$(MKLROOT)/lib/em64t $(MKLROOT)/lib/em64t/libmkl_lapack95_lp64.a -lmkl_intel_lp64 -lmkl_sequential -lmkl_core -lpthread -lm
```

Generation of parallel executable



Select Intel® product:	Intel(R) MKL 10.2
Select OS:	Linux*
Select compiler:	Intel(R) Fortran
Select architecture:	Intel(R) 64
Select dynamic or static linking:	Dynamic
Select interface layer:	LP64 (32-bit integer)
Select sequential or multi-threaded layer:	Multi-threaded
Select OpenMP library:	Intel(R) (libiomp5)
Select cluster library:	□ CDFT (BLACS required) □ ScaLAPACK (BLACS required) □ BLACS
Select MPI library:	<Select MPI>
Select the Fortran 95 interfaces:	□ BLAS95 ☑ LAPACK95
Link with Intel® MKL libraries explicitly:	□

Use this link line:

```
-L$(MKLROOT)/lib/em64t $(MKLROOT)/lib/em64t/libmkl_lapack95_lp64.a -
lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -lpthread -lm
```

Compiler options:

```
-openmp -I$(MKLROOT)/include/em64t/lp64 -I$(MKLROOT)/include
```

```
CFLAGS = -fast -openmp [redacted] -module $(OBJS_DIR) $(IDIR)
LFLAGS = -s
LIBS = -L$(MKLROOT)/lib/em64t $(MKLROOT)/lib/em64t/libmkl_lapack95_lp64.a -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -lpthread -lm -liomp5
```

References

- https://wiki.bath.ac.uk/display/HPC/OpenMP

- Intel® Math Kernel Library Reference Manual (http://software.intel.com/en-us/articles/intel-math-kernel-library-documentation)

- Intel® Math Kernel Library Link Line Advisor (http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor)

- Chapman, B., Jost, G., van der Pas, R., 2007. Using OpenMP: Portable Shared Memory Parallel Programming. MIT Press, Cambridge, Massachusetts, USA.

- www.openmp.org