HPL BENCHMARK

WEEK 2

Team Phoenix VIP, Fall 2025

HPL HANDS-ON LESSON #1

1. Additional overview of the High-Performance Linpack (HPL) Benchmark

2. Compile HPL on COC-ICE cluster

3. Perform test run to ensure successful compilation

4. Mini-assignment + goals for next week

HPL HANDS-ON LESSON #1

1. Additional overview of the High-Performance Linpack (HPL) Benchmark
2. Compile HPL on COC-ICE cluster
3. Perform test run to ensure successful compilation
4. Mini-assignment + goals for next week

WHAT IS HPL MEASURING?

- "Performance" is a vague term, so we need to be very exact at defining what our benchmark captures.
  - Reference point to describe specific features of your system, not a general measurement of its overall performance.

- **H**igh **P**erformance **L**INPACK (HPL) benchmark measures **fl**oating-point **o**perations **p**er **s**econd (FLOPS).

  - "…solves a (random) dense linear system in double precision (64 bits) arithmetic on distributed-memory computers" using row reduction.

WHY FLOPS?

- FLOPs is a standardized unit of measurement (Calahan, 1977) that shows computational throughput due to HPL's raw numerical computation (matrix operations).
- Hardware agnostic: can be measured on x86 and ARM CPU archs and GPUs.
- In HPL, reflects hardware performance and software optimization.
  - Hardware --> tests parallelism, cache, mem. hierarchy, and interconnects under computational load.
  - Software --> tests effectiveness of BLAS and MPI libs chosen.

Total GFLOPs = $2/3(N)^3 + 2(N)^2 / T_{solve} \times 10^9$

N = matrix size (NxN)

T_{solve} = total time (sec.) for completion

```
T/V                N      NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR11C2R4      125568     192     4     6            1184.68          1.1142e+03
```

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)	Rmax/Rpeak
1	**El Capitan** - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,039,616	1,742.00	2,746.38	29,581	63%
2	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607	65.8%
3	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698	51.1%
4	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84		66%
5	**HPC6** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, RHEL 8.9, HPE Eni S.p.A. Italy	3,143,520	477.90	606.97	8,461	78.7%

WHY FLOPS?

- Random dense matrix:
  - Easier to generate.
    - Sparse matrices may widely vary in sparsity patterns which affect memory access patterns and communication.
  - Can be applied across various architectures (e.g., CPUs & GPUs).
  - LA libs (e.g., BLAS, LAPACK) designed specifically for dense matrix operations. Allows for further performance tuning.

Dense Matrix

1	2	31	2	9	7	34	22	11	5
11	92	4	3	2	2	3	3	2	1
3	9	13	8	21	17	4	2	1	4
8	32	1	2	34	18	7	78	10	7
9	22	3	9	8	71	12	22	17	3
13	21	21	9	2	47	1	81	21	9
21	12	53	12	91	24	81	8	91	2
61	8	33	82	19	87	16	3	1	55
54	4	78	24	18	11	4	2	99	5
13	22	32	42	9	15	9	22	1	21

Sparse Matrix

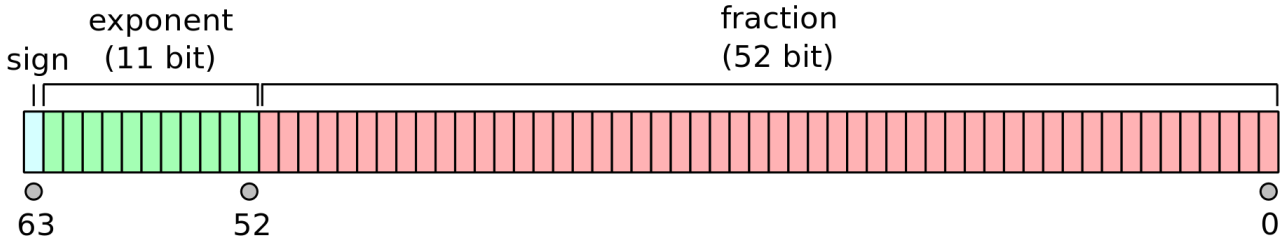
1	.	3	.	9	.	3	.	.	.
11	.	4	.	.	.	.	.	2	1
.	.	1	.	.	.	4	.	1	.
8	.	.	.	3	1	.	.	.	.
.	.	.	9	.	.	1	.	17	.
13	21	.	9	2	47	1	81	21	9
.	.	.	.	.	.	.	.	.	.
.	.	.	.	19	8	16	.	.	55
54	4	.	.	.	11	.	.	.	.
.	.	2	.	.	.	.	22	.	21

WHY FLOPS?

- "…solves a (random) dense linear system in double precision (64 bits) arithmetic on distributed-memory computers" using row reduction.

- Double precision (64-bit):
  - Compared to 32-bit…
    - Greater precision for fractional values.

IEEE 754 Format – 64-bit representation



Number → 0.1 or 1/10

Base-10, **0.1** is finite (non-repeating).
Base-2 (binary), **0.1** is infinite (repeating).

We need a method to approximate numbers that cannot be represented "accurately" in binary.

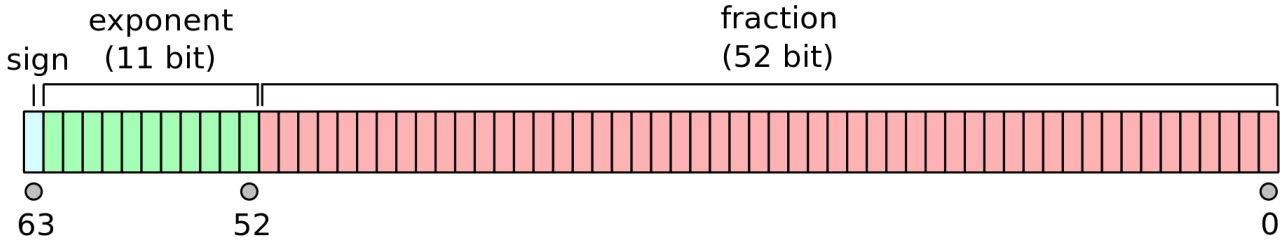
- Floating-point systems (e.g., 32- vs 64-bit)
- **0.1** in 32-bit = 0.100000001490116119384765625
  - Error ~ 1.49×10^{-8}
- **0.1** in 64-bit = 0.1000000000000000055511151231257827021181583404541015625
  - Error ~ 5.55×10^{-17}

WHY FLOPS?

- "…solves a (random) dense linear system in double precision (64 bits) arithmetic on distributed-memory computers" using row reduction.

- Double precision (64-bit):
  - Compared to 32-bit…
    - Greater precision for fractional values.
    - Increased range of numbers able to be represented.
      - Unsigned = 0 to 2^{64} vs 2^{32}

IEEE 754 Format – 64-bit representation



Number → 0.1 or 1/10

Base-10, **0.1** is finite (non-repeating).
Base-2 (binary), **0.1** is infinite (repeating).

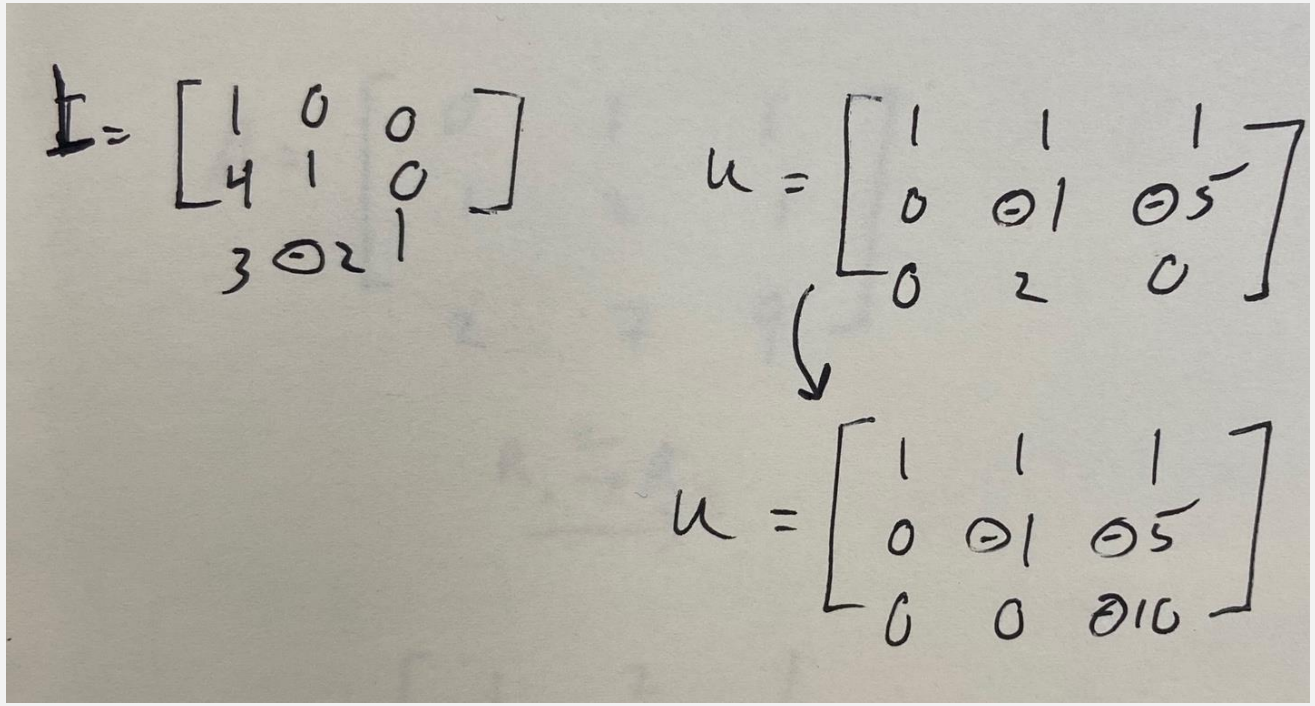
We need a method to approximate numbers that cannot be represented "accurately" in binary.

- Floating-point systems (e.g., 32- vs 64-bit)
- **0.1** in 32-bit = 0.100000001490116119384765625
  - Error ~ 1.49×10^{-8}
- **0.1** in 64-bit = 0.1000000000000000055511151231257827021181583404541015625
  - Error ~ 5.55×10^{-17}

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 4 & 3 & -1 \\ 3 & 5 & 3 \end{bmatrix}$$

Let's decompose this matrix and solve for L and U. (A = LU)

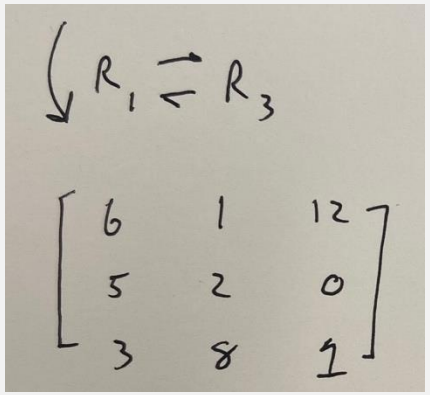
L

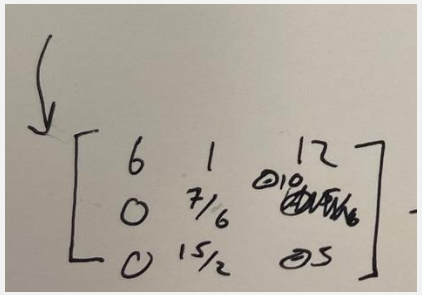


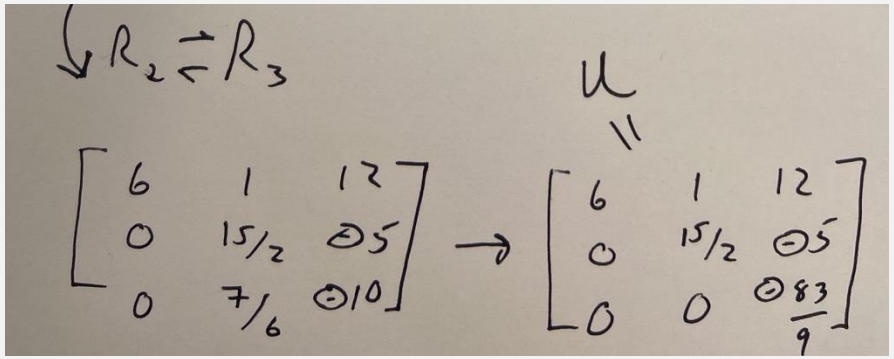
U

$$A = \begin{bmatrix} 3 & 8 & 1 \\ 5 & 2 & 0 \\ 6 & 1 & 12 \end{bmatrix}$$

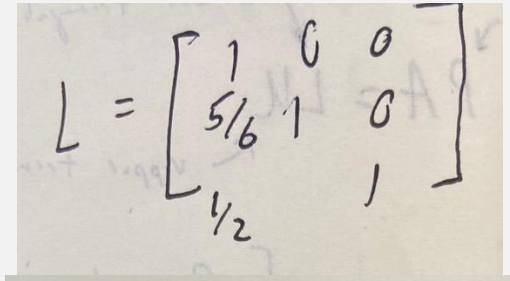
Let's decompose this matrix and solve for L and U, but this time via partial pivoting (PA = LU).

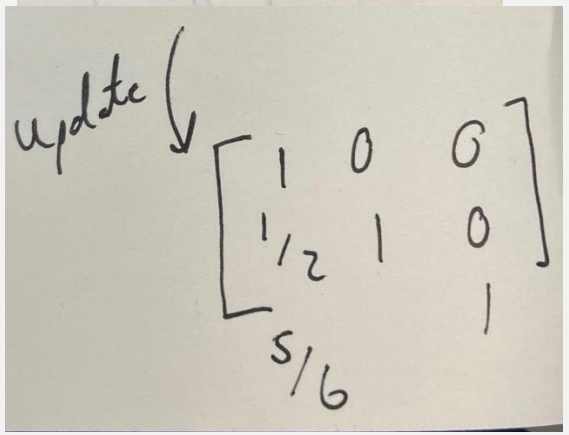


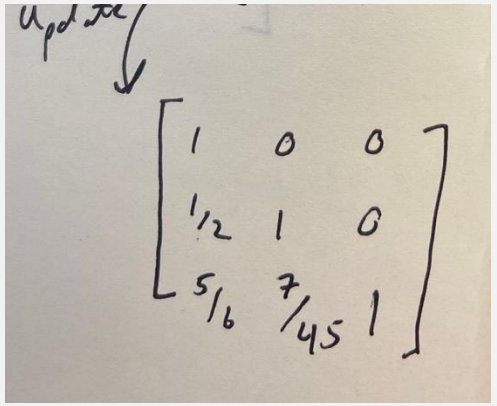




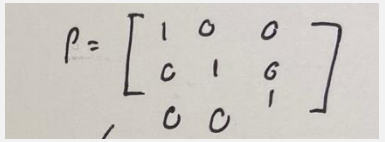
L triangular matrix:

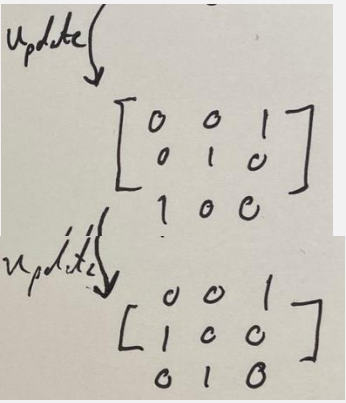






Permutation matrix:



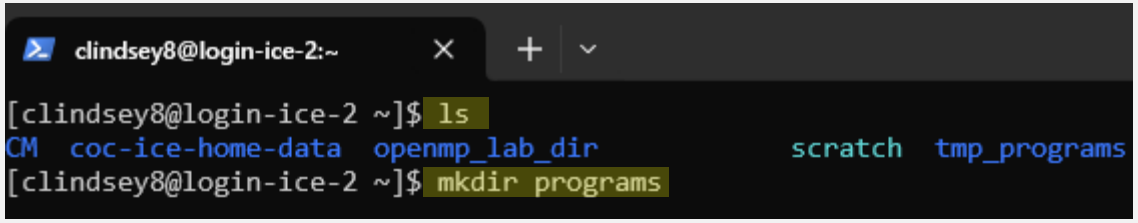


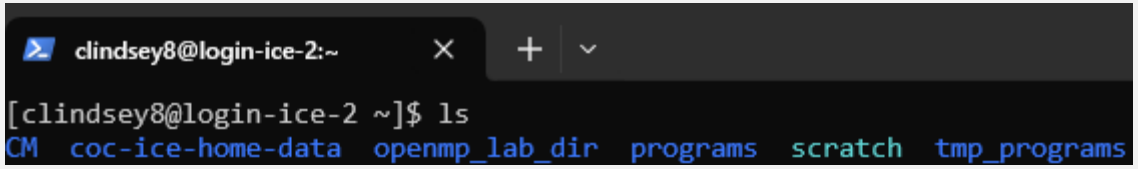
HPL HANDS-ON LESSON #1

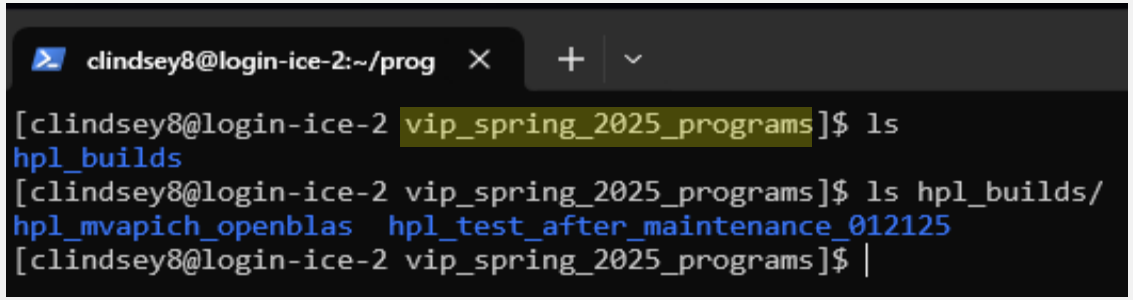
1. Additional overview of the High-Performance Linpack (HPL) Benchmark
2. Compile HPL on COC-ICE cluster
3. Perform test run to ensure successful compilation
4. Supplemental assignment + goals for next week

Let's SSH into the Instructional Cluster Environment (ICE)!

1.
```
ssh clindsey8@login-ice.pace.gatech.edu
```

2.


3.


4.


HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers

Innovative Computing Laboratory
UNIVERSITY OF TENNESSEE
COMPUTER SCIENCE DEPARTMENT

Version 2.3 | A. Petitet, R. C. Whaley, J. Dongarra, A. Cleary | December 2, 2018 | # Accesses

HPL is a software package that solves a (random) dense linear system in double precision (64 bits) arithmetic on distributed-memory computers. It can thus be regarded as a portable as well as freely available implementation of the High Performance Computing Linpack Benchmark.

The **algorithm** used by HPL can be summarized by the following keywords: Two-dimensional block-cyclic data distribution - Right-looking variant of the LU factorization with row partial pivoting featuring multiple look-ahead depths - Recursive panel factorization with pivot search and column broadcast combined - Various virtual panel broadcast topologies - bandwidth reducing swap-broadcast algorithm - backward substitution with look-ahead of depth 1.

The HPL package provides a testing and timing program to quantify the **accuracy** of the obtained solution as well as the time it took to compute it. The best **performance** achievable by this software on your system depends on a large variety of factors. Nonetheless, with some restrictive assumptions on the interconnection network, the algorithm described here and its attached implementation are **scalable** in the sense that their parallel efficiency is maintained constant with respect to the per processor memory usage.

The HPL software package **requires** the availibility on your system of an implementation of the Message Passing Interface **MPI** (1.1 compliant). An implementation of **either** the Basic Linear Algebra Subprograms **BLAS** **or** the Vector Signal Image Processing Library **VSIPL** is also needed. Machine-specific as well as generic implementations of MPI, the BLAS and VSIPL are available for a large variety of systems.

Acknowledgements: This work was supported in part by a grant from the Department of Energy's Lawrence Livermore National Laboratory and Los Alamos National Laboratory as part of the ASCI Projects contract numbers B503962 and 12187-001-00 4R.

Innovative Computing Laboratory
last revised December 2, 2018

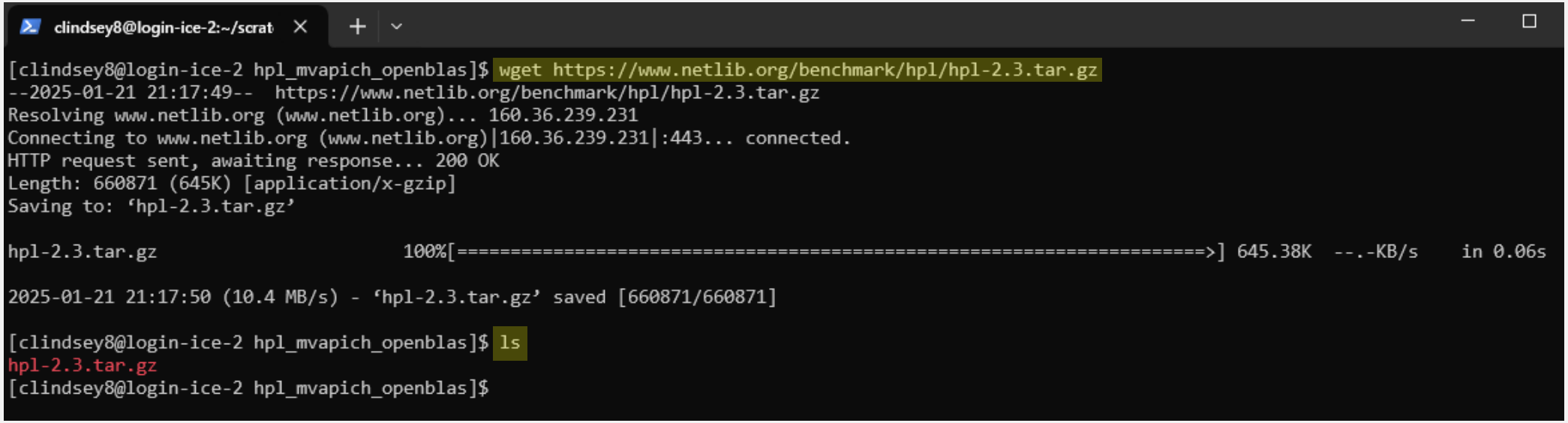
```
########################################################################

file    hpl-2.3.tar.gz
for     HPL 2.3 - A Portable Implementation of the High-Performance Linpack
,       Benchmark for Distributed-Memory Computers
by      Antoine Petitet, Clint Whaley, Jack Dongarra, Andy Cleary, Piotr Luszczek
Updated: December 2, 2018

########################################################################
```

https://www.netlib.org/benchmark/hpl/

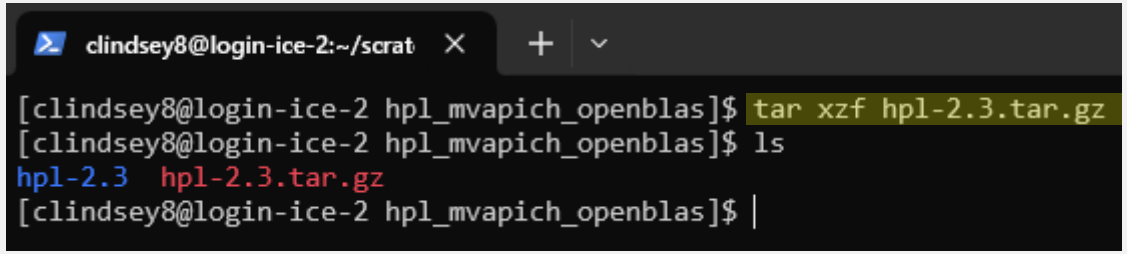
Google “netlib hpl”, and the first link should take you to the above page

Let's compile HPL on the Instructional Cluster Environment (ICE)!

5.

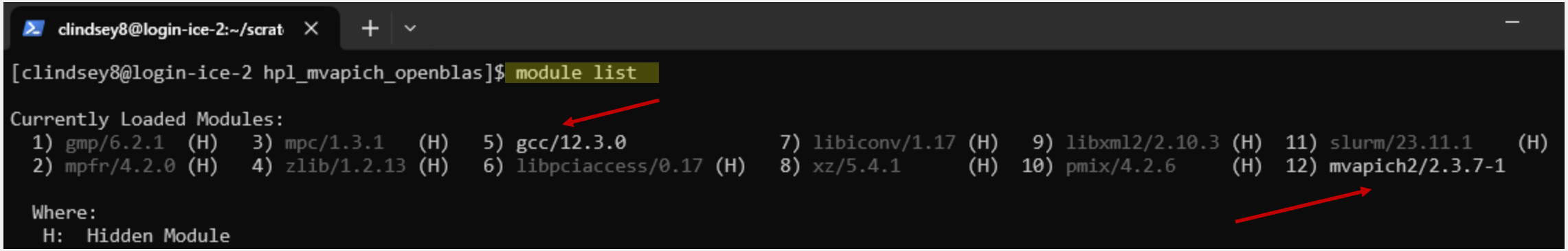


6.

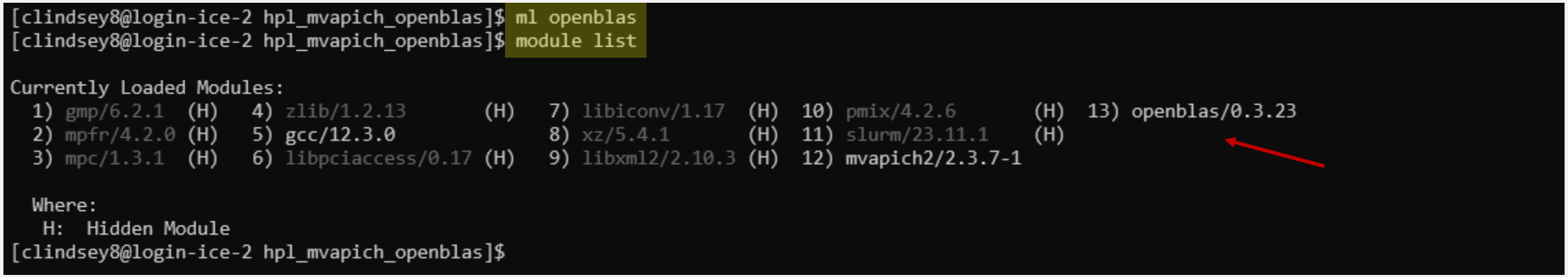


Let's compile HPL on the Instructional Cluster Environment (ICE)!

7.



8.



Let's compile HPL on the Instructional Cluster Environment (ICE)!

9.

```
[clindsey8@login-ice-2 hpl_mvapich_openblas]$ ls
hpl-2.3  hpl-2.3.tar.gz
[clindsey8@login-ice-2 hpl_mvapich_openblas]$ cd hpl-2.3/
[clindsey8@login-ice-2 hpl-2.3]$ ls
acinclude.m4  BUGS       config.guess  configure.ac  depcomp  INSTALL     Makefile.am  Make.top  NEWS    src      TODO
aclocal.m4    ChangeLog  config.sub    COPYING       HISTORY  install-sh  Makefile.in  man       README  testing  TUNING
AUTHORS       compile    configure     COPYRIGHT     include  Makefile    makes        missing   setup   THANKS   www
```

10.

```
[clindsey8@login-ice-2 hpl-2.3]$ cd setup/
[clindsey8@login-ice-2 setup]$ ls
Make.FreeBSD_PIV_CBLAS  Make.Linux_ATHLON_CBLAS  Make.Linux_PII_CBLAS_gm  Make.MacOSX_Accelerate  Make.SUN4SOL2-g_FBLAS  Make.UNKNOWN.in
make_generic            Make.Linux_ATHLON_FBLAS  Make.Linux_PII_FBLAS     Make.PWR2_FBLAS         Make.SUN4SOL2-g_VSIPL
Make.HPUX_FBLAS         Make.Linux_ATHLON_VSIPL  Make.Linux_PII_FBLAS_gm  Make.PWR3_FBLAS         Make.T3E_FBLAS
Make.I860_FBLAS         Make.Linux_Intel64       Make.Linux_PII_VSIPL     Make.PWRPC_FBLAS        Make.Tru64_FBLAS
Make.IRIX_FBLAS         Make.Linux_PII_CBLAS     Make.Linux_PII_VSIPL_gm  Make.SUN4SOL2_FBLAS     Make.Tru64_FBLAS_elan
```

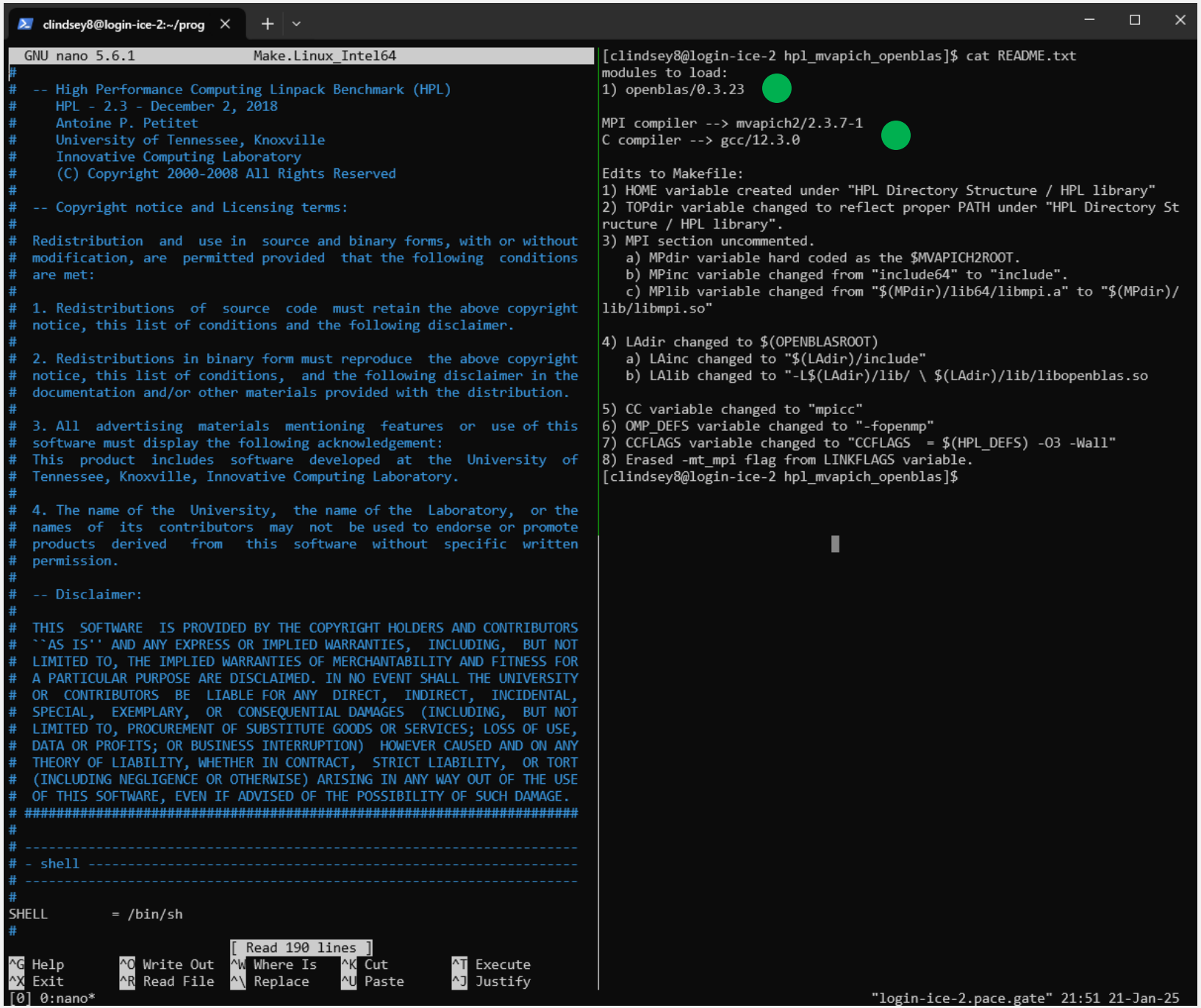
11.

```
[clindsey8@login-ice-2 setup]$ cp Make.Linux_Intel64 ..
[clindsey8@login-ice-2 setup]$
```

12.

```
[clindsey8@login-ice-2 setup]$ cd ..
[clindsey8@login-ice-2 hpl-2.3]$ ls
acinclude.m4  BUGS       config.guess  configure.ac  depcomp  INSTALL     Makefile.am         makes     missing  setup    THANKS  www
aclocal.m4    ChangeLog  config.sub    COPYING       HISTORY  install-sh  Makefile.in         Make.top  NEWS     src      TODO
AUTHORS       compile    configure     COPYRIGHT     include  Makefile    Make.Linux_Intel64  man       README   testing  TUNING
[clindsey8@login-ice-2 hpl-2.3]$ nano Make.Linux_Intel64
```

13.


clindsey8@login-ice-2:~/prog
GNU nano 5.6.1 Make.Linux_Intel64
-- High Performance Computing Linpack Benchmark (HPL)
HPL - 2.3 - December 2, 2018
Antoine P. Petitet
University of Tennessee, Knoxville
Innovative Computing Laboratory
(C) Copyright 2000-2008 All Rights Reserved
-- Copyright notice and Licensing terms:
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
# are met:
# 1. Redistributions of source code must retain the above copyright
# notice, this list of conditions and the following disclaimer.
# 2. Redistributions in binary form must reproduce the above copyright
# notice, this list of conditions, and the following disclaimer in the
# documentation and/or other materials provided with the distribution.
# 3. All advertising materials mentioning features or use of this
# software must display the following acknowledgement:
# This product includes software developed at the University of
# Tennessee, Knoxville, Innovative Computing Laboratory.
# 4. The name of the University, the name of the Laboratory, or the
# names of its contributors may not be used to endorse or promote
# products derived from this software without specific written
# permission.
# -- Disclaimer:
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
# ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
# A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE UNIVERSITY
# OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
# SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
# LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
# DATA OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
# THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
# (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
# OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
# ##
# --
# - shell --
# --
SHELL = /bin/sh
[Read 190 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify
[0] 0:nano*
[clindsey8@login-ice-2 hpl_mvapich_openblas]$ cat README.txt
modules to load:
1) openblas/0.3.23
MPI compiler --> mvapich2/2.3.7-1
C compiler --> gcc/12.3.0
Edits to Makefile:
1) HOME variable created under "HPL Directory Structure / HPL library"
2) TOPdir variable changed to reflect proper PATH under "HPL Directory Structure / HPL library".
3) MPI section uncommented.
a) MPdir variable hard coded as the $MVAPICH2ROOT.
b) MPinc variable changed from "include64" to "include".
c) MPlib variable changed from "$(MPdir)/lib64/libmpi.a" to "$(MPdir)/lib/libmpi.so"
4) LAdir changed to $(OPENBLASROOT)
a) LAinc changed to "$(LAdir)/include"
b) LAlib changed to "-L$(LAdir)/lib/ \ $(LAdir)/lib/libopenblas.so
5) CC variable changed to "mpicc"
6) OMP_DEFS variable changed to "-fopenmp"
7) CCFLAGS variable changed to "CCFLAGS = $(HPL_DEFS) -O3 -Wall"
8) Erased -mt_mpi flag from LINKFLAGS variable.
[clindsey8@login-ice-2 hpl_mvapich_openblas]$
"login-ice-2.pace.gate" 21:51 21-Jan-25

14.

clindsey8@login-ice-2:~/prog

```
  GNU nano 5.6.1                  Make.Linux_Intel64
# ######################################################################
#
# ----------------------------------------------------------------------
# - shell --------------------------------------------------------------
# ----------------------------------------------------------------------
#
SHELL        = /bin/sh
#
CD           = cd
CP           = cp
LN_S         = ln -fs
MKDIR        = mkdir -p
RM           = /bin/rm -f
TOUCH        = touch
#
# ----------------------------------------------------------------------
# - Platform identifier ------------------------------------------------
# ----------------------------------------------------------------------
#
ARCH         = Linux_Intel64
#
# ----------------------------------------------------------------------
# - HPL Directory Structure / HPL library ------------------------------
# ----------------------------------------------------------------------
#
HOME         = /home/hice1/clindsey8/programs/vip_spring_2025_programs/hp
TOPdir       = $(HOME)/hpl_mvapich_openblas/hpl-2.3
INCdir       = $(TOPdir)/include
BINdir       = $(TOPdir)/bin/$(ARCH)
LIBdir       = $(TOPdir)/lib/$(ARCH)
#
HPLlib       = $(LIBdir)/libhpl.a
#
# ----------------------------------------------------------------------
# - Message Passing library (MPI) --------------------------------------
# ----------------------------------------------------------------------
# MPinc tells the  C  compiler where to find the Message Passing library
# header files,  MPlib  is defined  to be the name of  the library to be
# used. The variable MPdir is only used for defining MPinc and MPlib.
#
```

```
[clindsey8@login-ice-2 hpl_mvapich_openblas]$ cat README.txt
modules to load:
1) openblas/0.3.23

MPI compiler --> mvapich2/2.3.7-1
C compiler --> gcc/12.3.0

Edits to Makefile:
1) HOME variable created under "HPL Directory Structure / HPL library"
2) TOPdir variable changed to reflect proper PATH under "HPL Directory St
ructure / HPL library".
3) MPI section uncommented.
   a) MPdir variable hard coded as the $MVAPICH2ROOT.
   b) MPinc variable changed from "include64" to "include".
   c) MPlib variable changed from "$(MPdir)/lib64/libmpi.a" to "$(MPdir)/
lib/libmpi.so"

4) LAdir changed to $(OPENBLASROOT)
   a) LAinc changed to "$(LAdir)/include"
   b) LAlib changed to "-L$(LAdir)/lib/ \ $(LAdir)/lib/libopenblas.so

5) CC variable changed to "mpicc"
6) OMP_DEFS variable changed to "-fopenmp"
7) CCFLAGS variable changed to "CCFLAGS  = $(HPL_DEFS) -O3 -Wall"
8) Erased -mt_mpi flag from LINKFLAGS variable.
[clindsey8@login-ice-2 hpl_mvapich_openblas]$
```

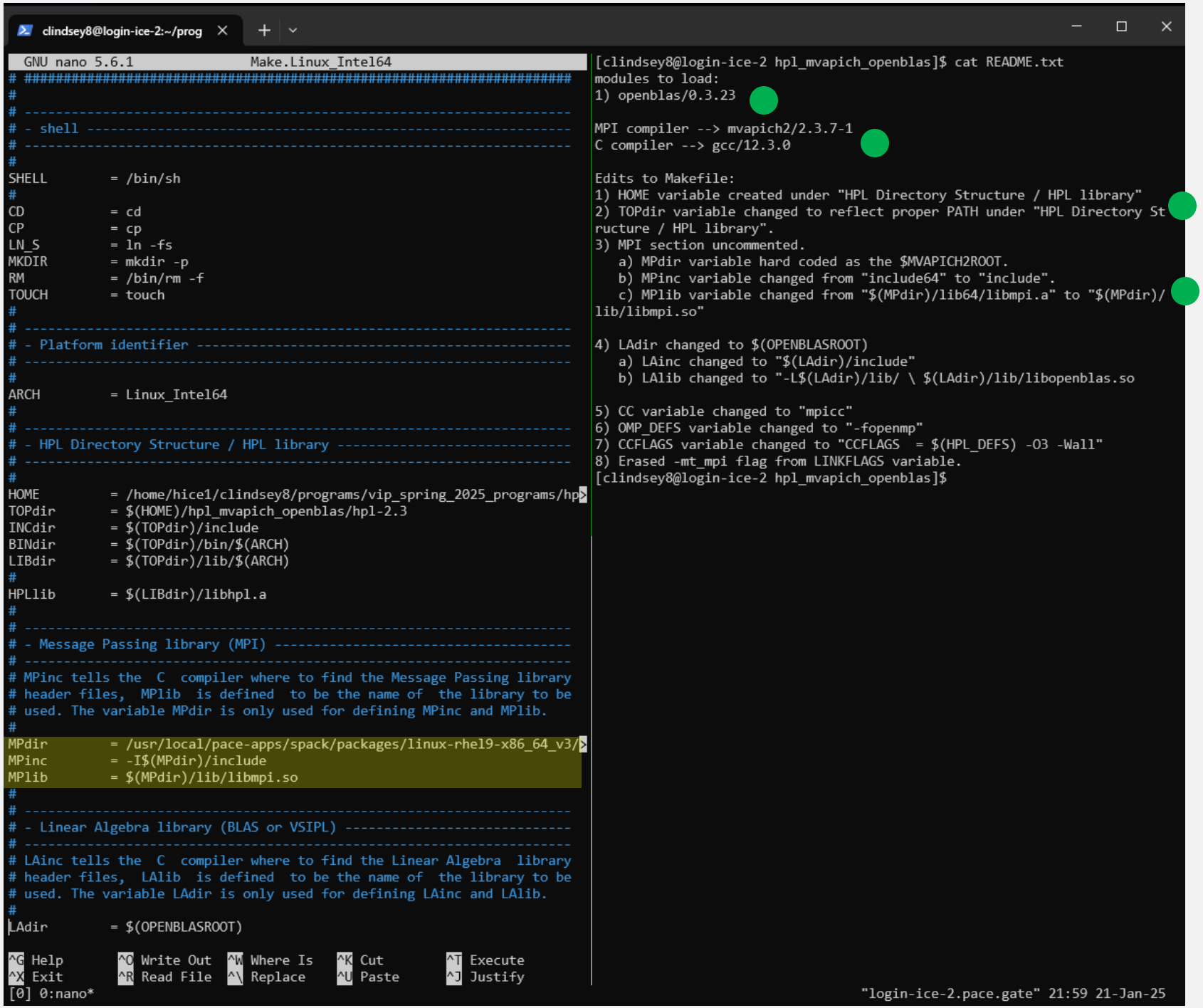
/home/hice1/clindsey8/programs/vip_spring_2025_programs/hpl_builds/hpl_mvapich_openblas/hpl-2.3

HOME var

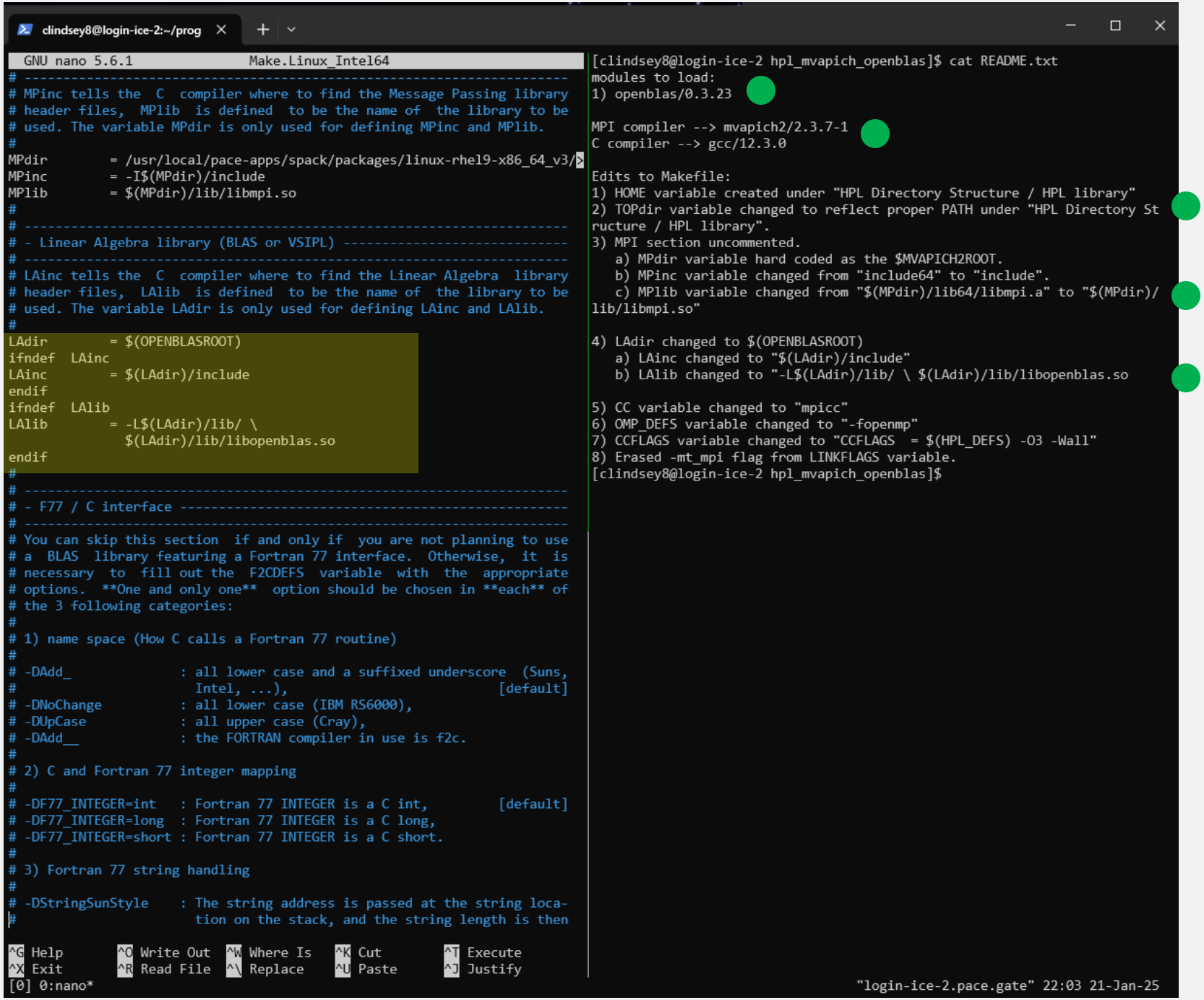
Do NOT include trailing slash in Home var

TOPdir var

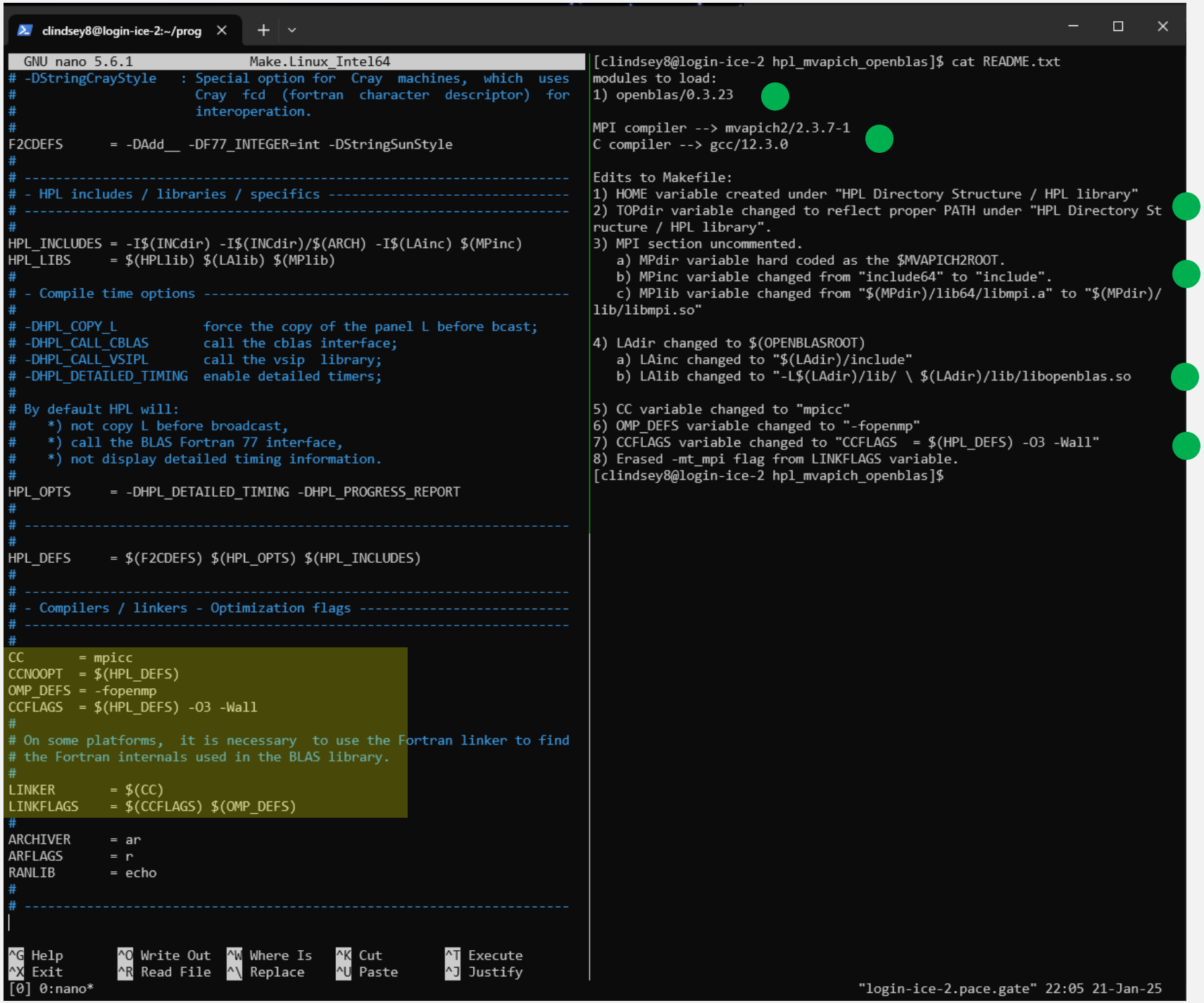
15.


clindsey8@login-ice-2:~/prog
GNU nano 5.6.1 Make.Linux_Intel64
# - shell ----------
SHELL = /bin/sh
CD = cd
CP = cp
LN_S = ln -fs
MKDIR = mkdir -p
RM = /bin/rm -f
TOUCH = touch
# - Platform identifier ----------
ARCH = Linux_Intel64
# - HPL Directory Structure / HPL library ----------
HOME = /home/hice1/clindsey8/programs/vip_spring_2025_programs/hp
TOPdir = $(HOME)/hpl_mvapich_openblas/hpl-2.3
INCdir = $(TOPdir)/include
BINdir = $(TOPdir)/bin/$(ARCH)
LIBdir = $(TOPdir)/lib/$(ARCH)
HPLlib = $(LIBdir)/libhpl.a
# - Message Passing library (MPI) ----------
# MPinc tells the C compiler where to find the Message Passing library
# header files, MPlib is defined to be the name of the library to be
# used. The variable MPdir is only used for defining MPinc and MPlib.
MPdir = /usr/local/pace-apps/spack/packages/linux-rhel9-x86_64_v3/
MPinc = -I$(MPdir)/include
MPlib = $(MPdir)/lib/libmpi.so
# - Linear Algebra library (BLAS or VSIPL) ----------
# LAinc tells the C compiler where to find the Linear Algebra library
# header files, LAlib is defined to be the name of the library to be
# used. The variable LAdir is only used for defining LAinc and LAlib.
LAdir = $(OPENBLASROOT)
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify
[0] 0:nano*
[clindsey8@login-ice-2 hpl_mvapich_openblas]$ cat README.txt
modules to load:
1) openblas/0.3.23
MPI compiler --> mvapich2/2.3.7-1
C compiler --> gcc/12.3.0
Edits to Makefile:
1) HOME variable created under "HPL Directory Structure / HPL library"
2) TOPdir variable changed to reflect proper PATH under "HPL Directory St
ructure / HPL library".
3) MPI section uncommented.
a) MPdir variable hard coded as the $MVAPICH2ROOT.
b) MPinc variable changed from "include64" to "include".
c) MPlib variable changed from "$(MPdir)/lib64/libmpi.a" to "$(MPdir)/
lib/libmpi.so"
4) LAdir changed to $(OPENBLASROOT)
a) LAinc changed to "$(LAdir)/include"
b) LAlib changed to "-L$(LAdir)/lib/ \ $(LAdir)/lib/libopenblas.so
5) CC variable changed to "mpicc"
6) OMP_DEFS variable changed to "-fopenmp"
7) CCFLAGS variable changed to "CCFLAGS = $(HPL_DEFS) -O3 -Wall"
8) Erased -mt_mpi flag from LINKFLAGS variable.
[clindsey8@login-ice-2 hpl_mvapich_openblas]$
"login-ice-2.pace.gate" 21:59 21-Jan-25

16.


clindsey8@login-ice-2:~/prog
GNU nano 5.6.1 Make.Linux_Intel64
MPdir = /usr/local/pace-apps/spack/packages/linux-rhel9-x86_64_v3/
MPinc = -I$(MPdir)/include
MPlib = $(MPdir)/lib/libmpi.so
LAdir = $(OPENBLASROOT)
ifndef LAinc
LAinc = $(LAdir)/include
endif
ifndef LAlib
LAlib = -L$(LAdir)/lib/ \
$(LAdir)/lib/libopenblas.so
endif
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify
[0] 0:nano*
[clindsey8@login-ice-2 hpl_mvapich_openblas]$ cat README.txt
modules to load:
1) openblas/0.3.23
MPI compiler --> mvapich2/2.3.7-1
C compiler --> gcc/12.3.0
Edits to Makefile:
1) HOME variable created under "HPL Directory Structure / HPL library"
2) TOPdir variable changed to reflect proper PATH under "HPL Directory Structure / HPL library".
3) MPI section uncommented.
a) MPdir variable hard coded as the $MVAPICH2ROOT.
b) MPinc variable changed from "include64" to "include".
c) MPlib variable changed from "$(MPdir)/lib64/libmpi.a" to "$(MPdir)/lib/libmpi.so"
4) LAdir changed to $(OPENBLASROOT)
a) LAinc changed to "$(LAdir)/include"
b) LAlib changed to "-L$(LAdir)/lib/ \ $(LAdir)/lib/libopenblas.so
5) CC variable changed to "mpicc"
6) OMP_DEFS variable changed to "-fopenmp"
7) CCFLAGS variable changed to "CCFLAGS = $(HPL_DEFS) -O3 -Wall"
8) Erased -mt_mpi flag from LINKFLAGS variable.
[clindsey8@login-ice-2 hpl_mvapich_openblas]$
"login-ice-2.pace.gate" 22:03 21-Jan-25

17.


clindsey8@login-ice-2:~/prog
GNU nano 5.6.1 Make.Linux_Intel64
-DStringCrayStyle : Special option for Cray machines, which uses
Cray fcd (fortran character descriptor) for
interoperation.
#
F2CDEFS = -DAdd__ -DF77_INTEGER=int -DStringSunStyle
#
--
- HPL includes / libraries / specifics -------------------------------
--
#
HPL_INCLUDES = -I$(INCdir) -I$(INCdir)/$(ARCH) -I$(LAinc) $(MPinc)
HPL_LIBS = $(HPLlib) $(LAlib) $(MPlib)
#
- Compile time options ---
#
-DHPL_COPY_L force the copy of the panel L before bcast;
-DHPL_CALL_CBLAS call the cblas interface;
-DHPL_CALL_VSIPL call the vsip library;
-DHPL_DETAILED_TIMING enable detailed timers;
#
By default HPL will:
*) not copy L before broadcast,
*) call the BLAS Fortran 77 interface,
*) not display detailed timing information.
#
HPL_OPTS = -DHPL_DETAILED_TIMING -DHPL_PROGRESS_REPORT
#
--
#
HPL_DEFS = $(F2CDEFS) $(HPL_OPTS) $(HPL_INCLUDES)
#
--
- Compilers / linkers - Optimization flags ---------------------------
--
#
CC = mpicc
CCNOOPT = $(HPL_DEFS)
OMP_DEFS = -fopenmp
CCFLAGS = $(HPL_DEFS) -O3 -Wall
#
On some platforms, it is necessary to use the Fortran linker to find
the Fortran internals used in the BLAS library.
#
LINKER = $(CC)
LINKFLAGS = $(CCFLAGS) $(OMP_DEFS)
#
ARCHIVER = ar
ARFLAGS = r
RANLIB = echo
#
--
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify
[0] 0:nano*
[clindsey8@login-ice-2 hpl_mvapich_openblas]$ cat README.txt
modules to load:
1) openblas/0.3.23
MPI compiler --> mvapich2/2.3.7-1
C compiler --> gcc/12.3.0
Edits to Makefile:
1) HOME variable created under "HPL Directory Structure / HPL library"
2) TOPdir variable changed to reflect proper PATH under "HPL Directory St
ructure / HPL library".
3) MPI section uncommented.
a) MPdir variable hard coded as the $MVAPICH2ROOT.
b) MPinc variable changed from "include64" to "include".
c) MPlib variable changed from "$(MPdir)/lib64/libmpi.a" to "$(MPdir)/
lib/libmpi.so"
4) LAdir changed to $(OPENBLASROOT)
a) LAinc changed to "$(LAdir)/include"
b) LAlib changed to "-L$(LAdir)/lib/ \ $(LAdir)/lib/libopenblas.so
5) CC variable changed to "mpicc"
6) OMP_DEFS variable changed to "-fopenmp"
7) CCFLAGS variable changed to "CCFLAGS = $(HPL_DEFS) -O3 -Wall"
8) Erased -mt_mpi flag from LINKFLAGS variable.
[clindsey8@login-ice-2 hpl_mvapich_openblas]$
"login-ice-2.pace.gate" 22:05 21-Jan-25

18.

```
clindsey8@login-ice-2:~/prog
[clindsey8@login-ice-2 hpl_mvapich_openblas]$ salloc -N1 -n8 -t15:00
salloc: Pending job allocation 1238919
salloc: job 1238919 queued and waiting for resources
salloc: job 1238919 has been allocated resources
salloc: Granted job allocation 1238919
salloc: Waiting for resource configuration
salloc: Nodes atl1-1-02-003-20-1 are ready for job
----------------------------------------
Begin Slurm Prolog: Jan-21-2025 22:10:07
Job ID:    1238919
User ID:   clindsey8
Account:   bios
Job name:  interactive
Partition: coc-cpu,ice-cpu
QOS:       coc-ice
----------------------------------------
```

19.

```
[clindsey8@atl1-1-02-003-20-1 hpl-2.3]$ cd bin/Linux_Intel64/
[clindsey8@atl1-1-02-003-20-1 Linux_Intel64]$ ls
HPL.dat  xhpl
[clindsey8@atl1-1-02-003-20-1 Linux_Intel64]$ srun xhpl HPL.dat
```

20.

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR00R2R4          35     4     4     1               0.00            5.3769e-02
HPL_pdgesv() start time Tue Jan 21 22:10:56 2025

HPL_pdgesv() end time   Tue Jan 21 22:10:56 2025

--VV--VV--VV--VV--VV--VV--VV--VV--VV--VV--VV--VV--VV--VV--VV--VV--VV-
Max aggregated wall time rfact . . . :               0.00
+ Max aggregated wall time pfact . . :               0.00
+ Max aggregated wall time mxswp . . :               0.00
Max aggregated wall time update  . . :               0.00
+ Max aggregated wall time laswp . . :               0.00
Max aggregated wall time up tr sv  . :               0.00
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   2.22409445e-02 ..... PASSED
================================================================================

Finished    864 tests with the following results:
            864 tests completed and passed residual checks,
              0 tests completed and failed residual checks,
              0 tests skipped because of illegal input values.
--------------------------------------------------------------------------------

End of Tests.
================================================================================
[clindsey8@atl1-1-02-003-20-1 Linux_Intel64]$
[0] 0:srun*
```

memes
sc21
isc22
sc22
isc23
sc23
memxct
sp24mp1
sc24
indyscc24

APP SPECIFIC

hpl
io500
hpcg
mystery
ramble

November 13, 2024

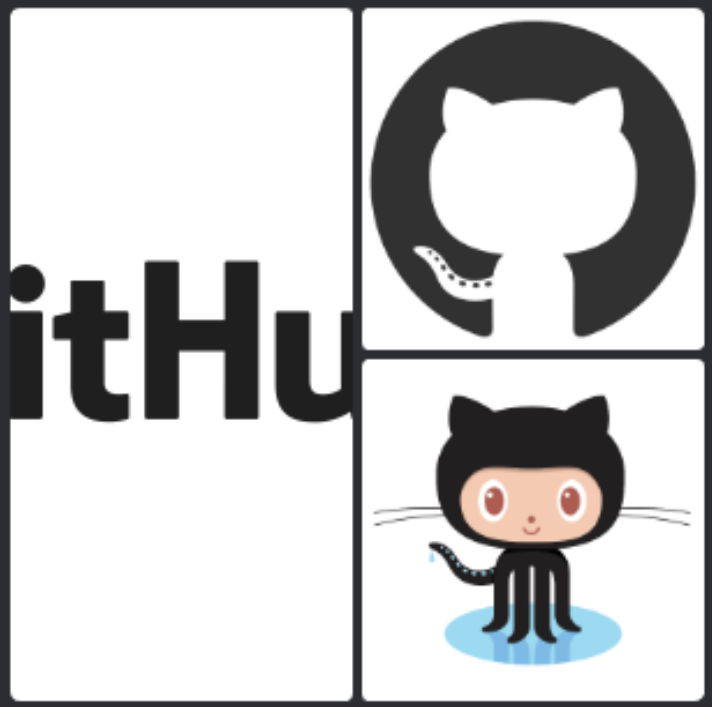
Colonl_Charisma 11/13/2024 3:38

https://github.gatech.edu/clindsey8/HPL_VIP_Module

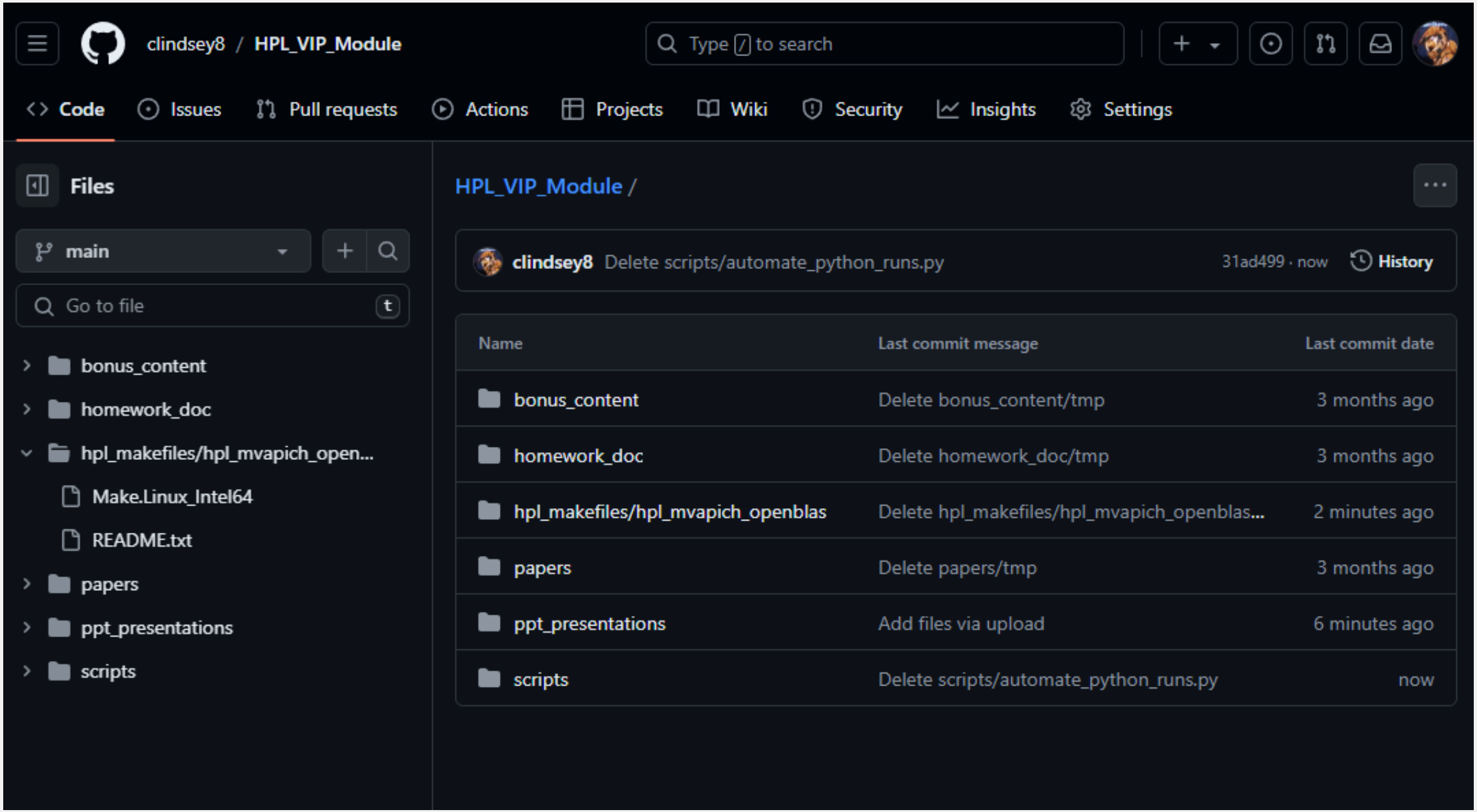
GitHub

Build software better, together

GitHub is where people build software. More than 100 million people use GitHub to discover, fork, and contribute to over 420 million projects.



Message #hpl


clindsey8 / HPL_VIP_Module
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Files
main
Go to file
t
bonus_content
homework_doc
hpl_makefiles/hpl_mvapich_open...
Make.Linux_Intel64
README.txt
papers
ppt_presentations
scripts
HPL_VIP_Module /
clindsey8 Delete scripts/automate_python_runs.py
31ad499 · now
History
Name
Last commit message
Last commit date
bonus_content
Delete bonus_content/tmp
3 months ago
homework_doc
Delete homework_doc/tmp
3 months ago
hpl_makefiles/hpl_mvapich_openblas
Delete hpl_makefiles/hpl_mvapich_openblas...
2 minutes ago
papers
Delete papers/tmp
3 months ago
ppt_presentations
Add files via upload
6 minutes ago
scripts
Delete scripts/automate_python_runs.py
now

HPL HANDS-ON LESSON #1

1. Additional overview of the High-Performance Linpack (HPL) Benchmark
2. Compile HPL on COC-ICE cluster
3. Perform test run to ensure successful compilation
4. Mini-assignment + goals for next week

Mini-assignment

```
            HPL_DAT_FILE = f'''HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6            device out (6=stdout,7=stderr,file)
1            # of problems sizes (N)
{Ns_DIVISIBLE_BY_NB}         Ns
1            # of NBs
{NB}           NBs
0            PMAP process mapping (0=Row-,1=Column-major)
1            # of process grids (P x Q)
{pq[0]}            Ps
{pq[1]}            Qs
16.0         threshold
1            # of panel fact
2            PFACTs (0=left, 1=Crout, 2=Right)
1            # of recursive stopping criterium
4            NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
1            # of recursive panel fact.
2            RFACTs (0=left, 1=Crout, 2=Right)
1            # of broadcast
1            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1            # of lookahead depth
1            DEPTHs (>=0)
2            SWAP (0=bin-exch,1=long,2=mix)
64           swapping threshold
0            L1 in (0=transposed,1=no-transposed) form
0            U  in (0=transposed,1=no-transposed) form
1            Equilibration (0=no,1=yes)
{mem_aln}            memory alignment in double (> 0)'''
```

HPL Input (HPL.dat)

Important parameters:

- N #Matrix dimension
- NB #Block size for LA operations
- P #Factorization rows
- Q #Factorization columns. PxQ must equal your MPI processes
- Change # of algorithms to test

Run HPL for 1, 2 and 4 nodes, tune the HPL.dat file to achieve the best GFLOPs number.

- Which parameters did you tune and why?
- Plot the GFLOPs number for each run vs. no. of cores.
- Was the GFLOPs score of the 2-node run exactly twice that of your single-node performance? Why or why not?