### 3.2. Linpack on a Single AWS Node (15 points)

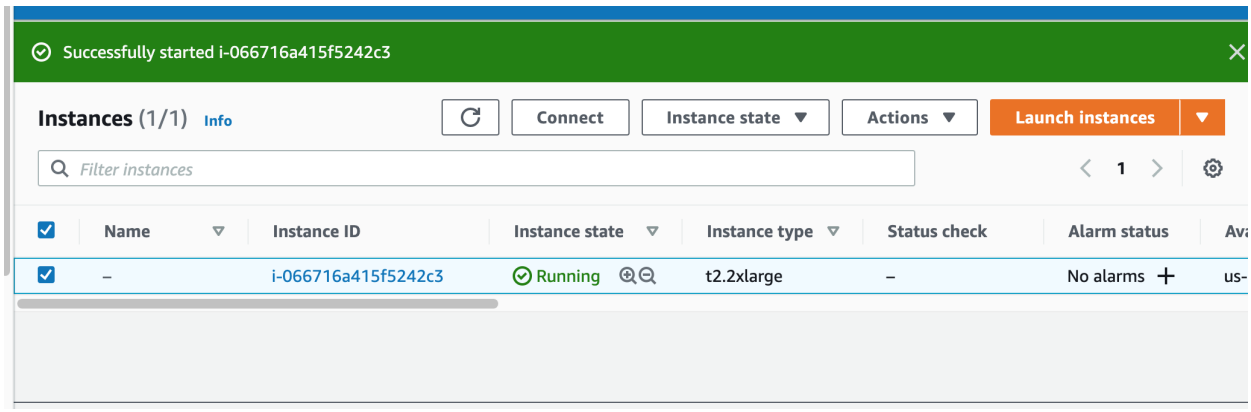
Download[1], compile, and run the Linpack benchmark with **4 processes on one t2.2xlarge instance** on AWS running Ubuntu. You may need to install MPI in the node, see the course Infrastructure Guide "MPI on AWS". Tune the execution of the benchmark to maximize performance. Discuss and compare the performance with the historical results of the benchmark.

**Submission**

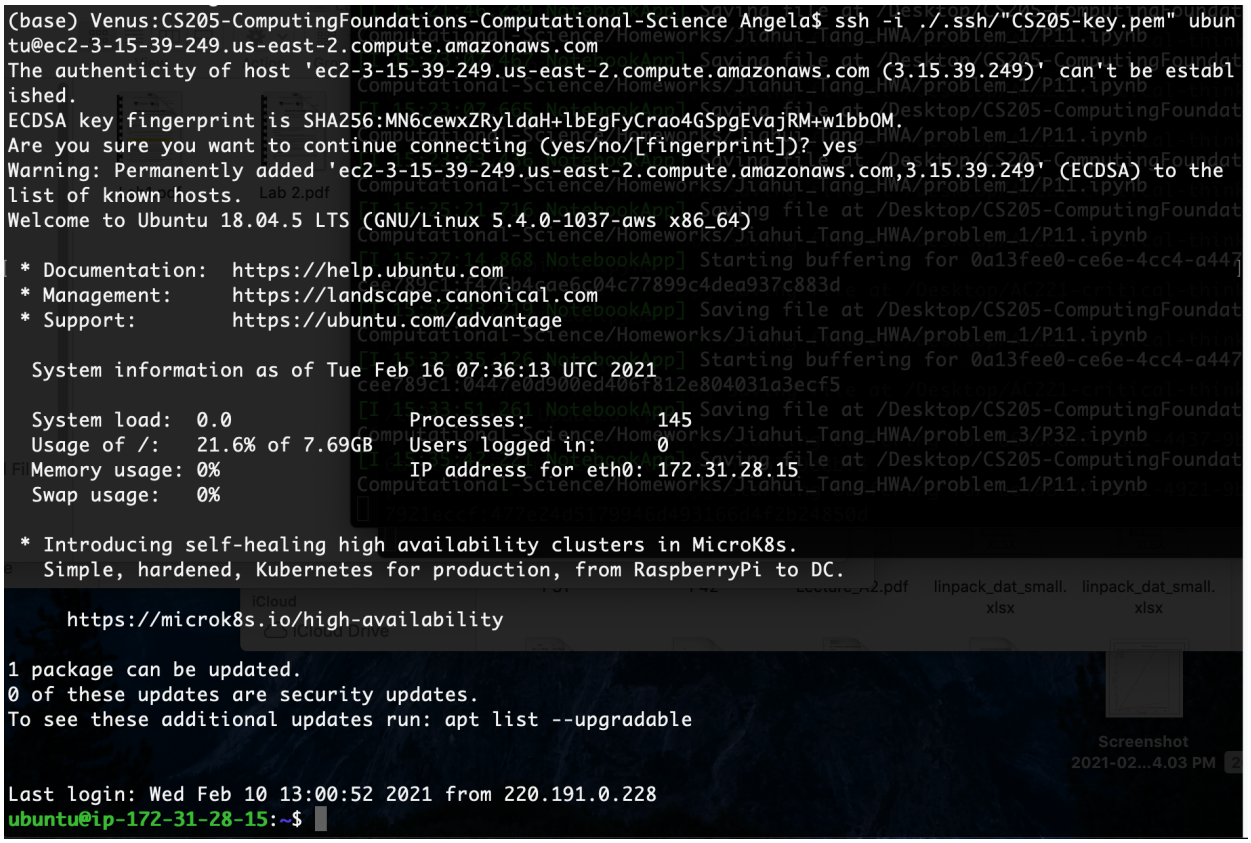
- `P32.pdf`: Explain steps and configuration needed to compile and run the benchmark (i.e. high level details such as what instance of AWS and what commands used for compiling), explain the parameters tuned to maximize performance (i.e. report the maximum Gflop), represent the obtained performance in the Top500 performance development graph[2]

## By Jiahui Tang

STEP 1. Start the t2.2xlarge Instance




STEP 2. SSH into my Ubuntu Instance



```
(base) Venus:CS205-ComputingFoundations-Computational-Science Angela$ ssh -i ./.ssh/"CS205-key.pem" ubun
tu@ec2-3-15-39-249.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-15-39-249.us-east-2.compute.amazonaws.com (3.15.39.249)' can't be establ
ished.
ECDSA key fingerprint is SHA256:MN6cewxZRyldaH+lbEgFyCrao4GSpgEvajRM+w1bbOM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-15-39-249.us-east-2.compute.amazonaws.com,3.15.39.249' (ECDSA) to the
list of known hosts.
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1037-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Feb 16 07:36:13 UTC 2021

  System load:  0.0               Processes:           145
  Usage of /:   21.6% of 7.69GB   Users logged in:     0
  Memory usage: 0%                IP address for eth0: 172.31.28.15
  Swap usage:   0%

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

1 package can be updated.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


Last login: Wed Feb 10 13:00:52 2021 from 220.191.0.228
ubuntu@ip-172-31-28-15:~$
```

STEP 3. Follow the Guide to install linpack package.

```
ubuntu@ip-172-31-28-15:~$ ls
hpl-2.3  hpl-2.3.tar.gz  mpi_sc  mpi_sc.c
ubuntu@ip-172-31-28-15:~$ mv hpl-2.3 hpl
ubuntu@ip-172-31-28-15:~$ ls
hpl  hpl-2.3.tar.gz  mpi_sc  mpi_sc.c
ubuntu@ip-172-31-28-15:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu bionic-security InRelease
Reading package lists... Done
ubuntu@ip-172-31-28-15:~$ sudo apt-get install make
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  make-doc
The following NEW packages will be installed:
  make
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 154 kB of archives.
After this operation, 381 kB of additional disk space will be used.
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 make amd64 4.1-9.1ubuntu1 [154 kB
]
Fetched 154 kB in 0s (0 B/s)
Selecting previously unselected package make.
(Reading database ... 63145 files and directories currently installed.)
Preparing to unpack .../make_4.1-9.1ubuntu1_amd64.deb ...
Unpacking make (4.1-9.1ubuntu1) ...
Setting up make (4.1-9.1ubuntu1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
ubuntu@ip-172-31-28-15:~$
```

STEP 4. Edit the /etc/hosts file and Copy Internal Host name and IP Address

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Ava |
|---|---|---|---|---|---|---|
| – | i-066716a415f5242c3 | Running | t2.2xlarge | 2/2 checks ... | No alarms + | us- |

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

**Networking details** Info

| Public IPv4 address | Private IPv4 addresses | VPC ID |
|---|---|---|
| 3.15.39.249 \| open address | 172.31.28.15 | vpc-7cb83f17 |
| **Public IPv4 DNS** | **Private IPv4 DNS** | **Subnet ID** |
| ec2-3-15-39-249.us-east-2.compute.amazonaws.com \| open address | ip-172-31-28-15.us-east-2.compute.internal | subnet-6e43a613 |
| **IPV6 addresses** | **Secondary private IPv4 addresses** | **Availability zone** |
| – | – | us-east-2b |

```
127.0.0.1 localhost
172.31.28.15 ip-172-31-28-15.us-east-2.compute.internal

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
~
~
~
~
~
~
~
~
~
~
~
~
~
```

STEP 5. Follow the guide to Make.UNKNOWN to the top directory and install hpl, to use MPI run the hpl

```
----------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   3.57360298e-02 ...... PASSED
============================================================================
T/V                N    NB     P     Q               Time                 Gflops
----------------------------------------------------------------------------
WR00R2R2          35     4     4     1               0.00              3.0452e-01
HPL_pdgesv() start time Tue Feb 16 07:51:53 2021

HPL_pdgesv() end time   Tue Feb 16 07:51:53 2021

----------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   2.07165390e-02 ...... PASSED
============================================================================
T/V                N    NB     P     Q               Time                 Gflops
----------------------------------------------------------------------------
WR00R2R4          35     4     4     1               0.00              3.1505e-01
HPL_pdgesv() start time Tue Feb 16 07:51:53 2021

HPL_pdgesv() end time   Tue Feb 16 07:51:53 2021

----------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   2.07165390e-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.
============================================================================
```

STEP 6. OPEN TUNING FILE to learn about how to tune parameters

```
ubuntu@ip-172-31-28-15:~/hpl$ cat TUNING
================================================================
Performance Tuning and setting up the input data file HPL.dat

Current as of release HPL - 2.3 - December 2, 2018
================================================================
Check out  the website  www.netlib.org/benchmark/hpl  for the
latest information.

After  having  built  the executable hpl/bin/<arch>/xhpl, one
may want to modify the input  data  file  HPL.dat.  This file
should  reside  in  the  same  directory  as  the  executable
hpl/bin/<arch>/xhpl.  An example  HPL.dat file is provided by
default.  This  file  contains  information about the problem
sizes,  machine configuration,  and  algorithm features to be
used by the executable. It is 30 lines long. All the selected
parameters  will  be  printed  in the output generated by the
executable.

At the end of this file,  there  is a couple of  experimental
guide lines that you may find useful.

================================================================
File HPL.dat (description):

Line 1: (unused) Typically  one  would  use this line for its
own good. For example, it could be used to summarize the con-
tent of the input file. By default this line reads:

HPL Linpack benchmark input file

Line 2: (unused) same as line 1. By default this line reads:

Innovative Computing Laboratory, University of Tennessee
```

STEP 7. TUNING hpl.dat to below parameters and rerun

```
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)
5             # of problems sizes (N)
1024 2048 4096 8192 16384 Ns
12            # of NBs
80 100 144 20 30 40 50 60 70 80 90 101      NBs
0             PMAP process mapping (0=Row-,1=Column-major)
2             # of process grids (P x Q)
2 4         Ps
2 1         Qs
16.0          threshold
1             # of panel fact
1         PFACTs (0=left, 1=Crout, 2=Right)
2             # of recursive stopping criterium
4 8           NBMINs (>= 1)
1             # of panels in recursion
2             NDIVs
1             # of recursive panel fact.
1         RFACTs (0=left, 1=Crout, 2=Right)
2             # of broadcast
1 3             BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
2             # of lookahead depth
0 1             DEPTHs (>=0)
2             SWAP (0=bin-exch,1=long,2=mix)
60            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)
8             memory alignment in double (> 0)
~
```

STEP 8. add -fast flag to CCFLAGS, change CC = mpiicc to use icc instead of gcc, and re-compile make.UNKNOWN

Note: tried with https://software.intel.com/content/www/us/en/develop/articles/installing-intel-oneapi-toolkits-via-apt.html but it seems not work (trying to change compiler from gcc to icc and add optimization flag)

```
# ----------------------------------------------------------------------
#
HPL_DEFS     = $(F2CDEFS) $(HPL_OPTS) $(HPL_INCLUDES)
#
# ----------------------------------------------------------------------
# - Compilers / linkers - Optimization flags ---------------------------
# ----------------------------------------------------------------------
#
CC           = mpiicc
CCNOOPT      = $(HPL_DEFS)
CCFLAGS      = $(HPL_DEFS) -fast
#
LINKER       = mpif77
LINKFLAGS    =
#
ARCHIVER     = ar
ARFLAGS      = r
RANLIB       = echo
#
# ----------------------------------------------------------------------
-- INSERT --
```

STEP 9. When N gets larger, PxQ is squared grid, it acheive good performance, where the best is around **36 GFlops/sec**

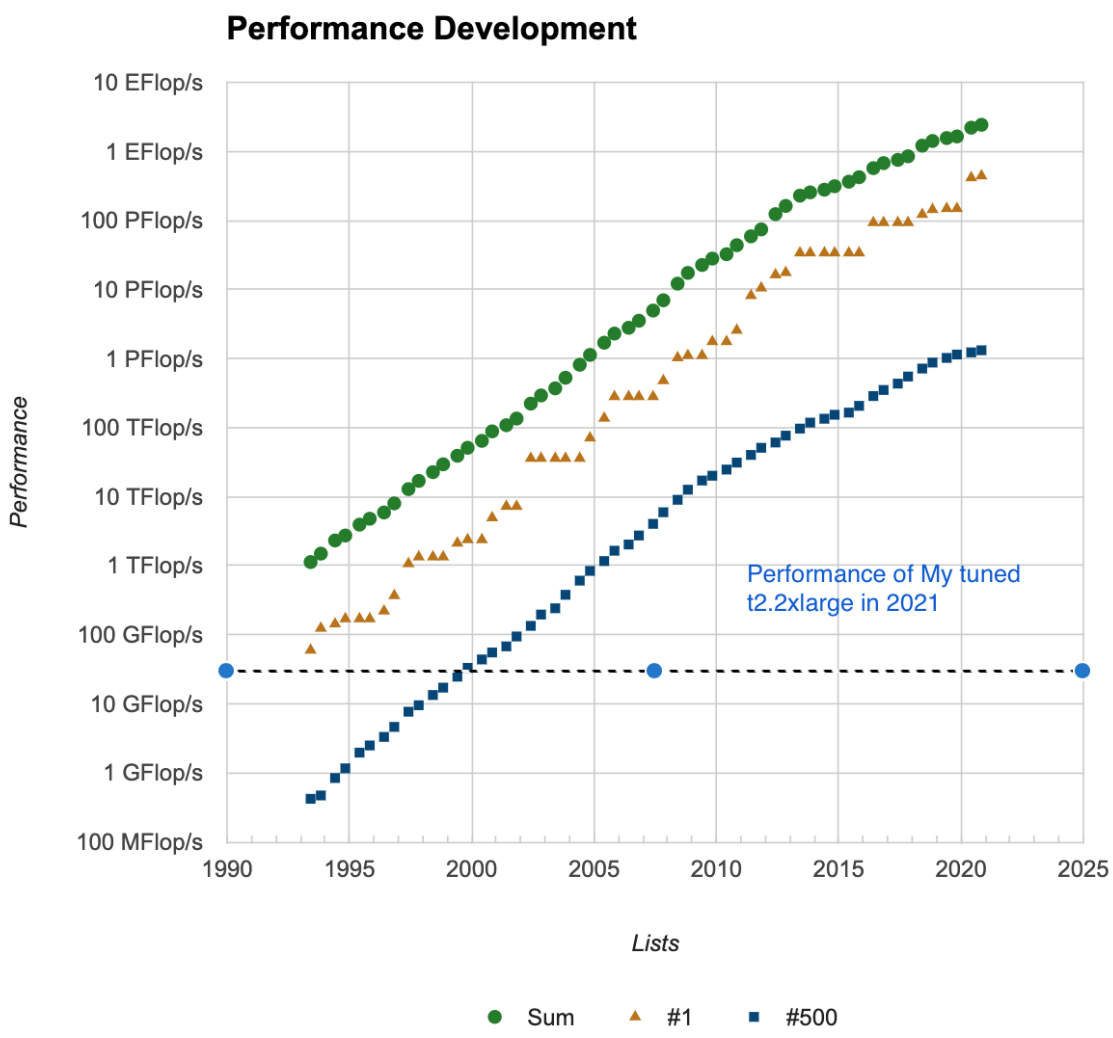
```
----------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   1.43895085e-03 ...... PASSED
============================================================================
T/V                N    NB     P     Q               Time                 Gflops
----------------------------------------------------------------------------
WR11C2C4        8192    80     2     2              10.64              3.4458e+01
HPL_pdgesv() start time Tue Feb 16 10:10:28 2021

HPL_pdgesv() end time   Tue Feb 16 10:10:39 2021

----------------------------------------------------------------------------
```

```
----------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   1.53739848e-03 ...... PASSED
============================================================================
T/V                N    NB     P     Q               Time                 Gflops
----------------------------------------------------------------------------
WR01C2C4        8192   144     2     2              10.60              3.4589e+01
HPL_pdgesv() start time Tue Feb 16 10:13:30 2021

HPL_pdgesv() end time   Tue Feb 16 10:13:40 2021
```

```
HPL_pdgesv() end time   Tue Feb 16 12:32:58 2021

----------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   1.03049124e-03 ...... PASSED
============================================================================
T/V                N    NB     P     Q               Time                 Gflops
----------------------------------------------------------------------------
WR01C2C8       16384    70     2     2              82.24              3.5656e+01
HPL_pdgesv() start time Tue Feb 16 12:32:56 2021

HPL_pdgesv() end time   Tue Feb 16 12:34:18 2021

----------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   1.04206321e-03 ...... PASSED
============================================================================
T/V                N    NB     P     Q               Time                 Gflops
----------------------------------------------------------------------------
WR03C2C4       16384    70     2     2              82.26              3.5647e+01
HPL_pdgesv() start time Tue Feb 16 12:34:35 2021

HPL_pdgesv() end time   Tue Feb 16 12:35:57 2021

----------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   1.03049124e-03 ...... PASSED
============================================================================
T/V                N    NB     P     Q               Time                 Gflops
----------------------------------------------------------------------------
WR03C2C8       16384    70     2     2              82.28              3.5641e+01
HPL_pdgesv() start time Tue Feb 16 12:36:14 2021

HPL_pdgesv() end time   Tue Feb 16 12:37:37 2021
```

```
----------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=   1.05253658e-03 ...... PASSED
============================================================================
T/V                N    NB     P     Q               Time                 Gflops
----------------------------------------------------------------------------
WR13C2C4       16384   100     2     2              79.96              3.6675e+01
HPL_pdgesv() start time Tue Feb 16 10:58:12 2021

HPL_pdgesv() end time   Tue Feb 16 10:59:32 2021
```

Obtained Performanced Graph:


Performance Development
10 EFlop/s
1 EFlop/s
100 PFlop/s
10 PFlop/s
1 PFlop/s
100 TFlop/s
10 TFlop/s
1 TFlop/s
100 GFlop/s
10 GFlop/s
1 GFlop/s
100 MFlop/s
Performance
1990
1995
2000
2005
2010
2015
2020
2025
Lists
Performance of My tuned
t2.2xlarge in 2021
Sum
#1
#500

```
In [ ]:
```