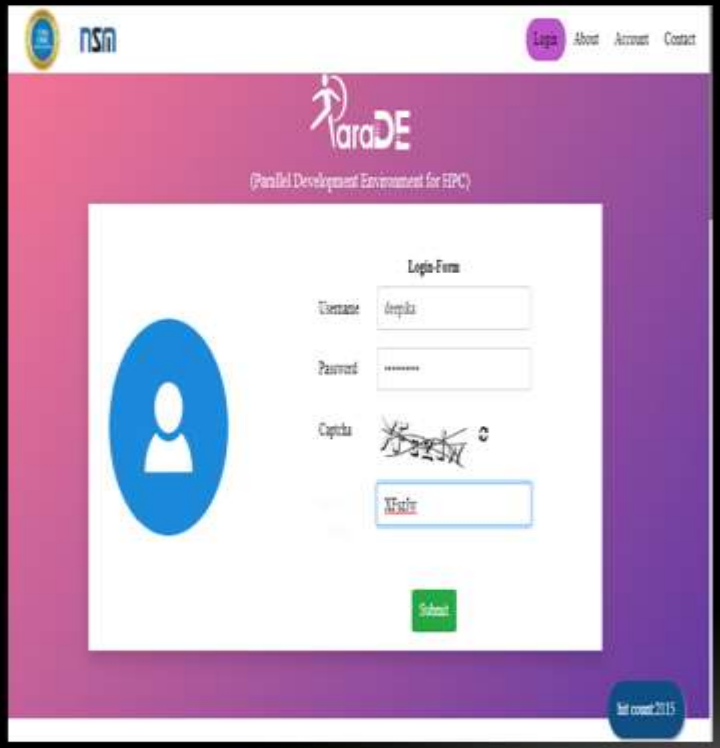
PARADE

Parallel Development Environment for HPC

PARADE

- Integrated Development Environment for HPC



- Browser based IDE
- No installation required on the client machine
- Single interface for application development

WORKING ENVIRONMENT IN HPC

Using Command-line !!

Login to the HPC Cluster

```
parade@login03:~
login as: parade
Using keyboard-interactive authentication.
If you truly desire access to this host, then you must indulge me in a simple challenge.
------------------------------------------------------------------


Observe the picture below and answer the question listed afterwards:

 _ _ _ _ _ _ _ _ _
/\ /\ /\ /\ /\ /\ /\ /\ /\
( A | k | x | Q | E | f | O | n )
 \/ \/ \/ \/ \/ \/ \/ \/ \/

Type the string above: AkxQEfOn
Using keyboard-interactive authentication.
Password:
Last login: Wed Apr 20 17:43:58 2022 from deepika-workstation.blr.cdac.in
[parade@login03 ~]$ vi pi-calulation.c
```

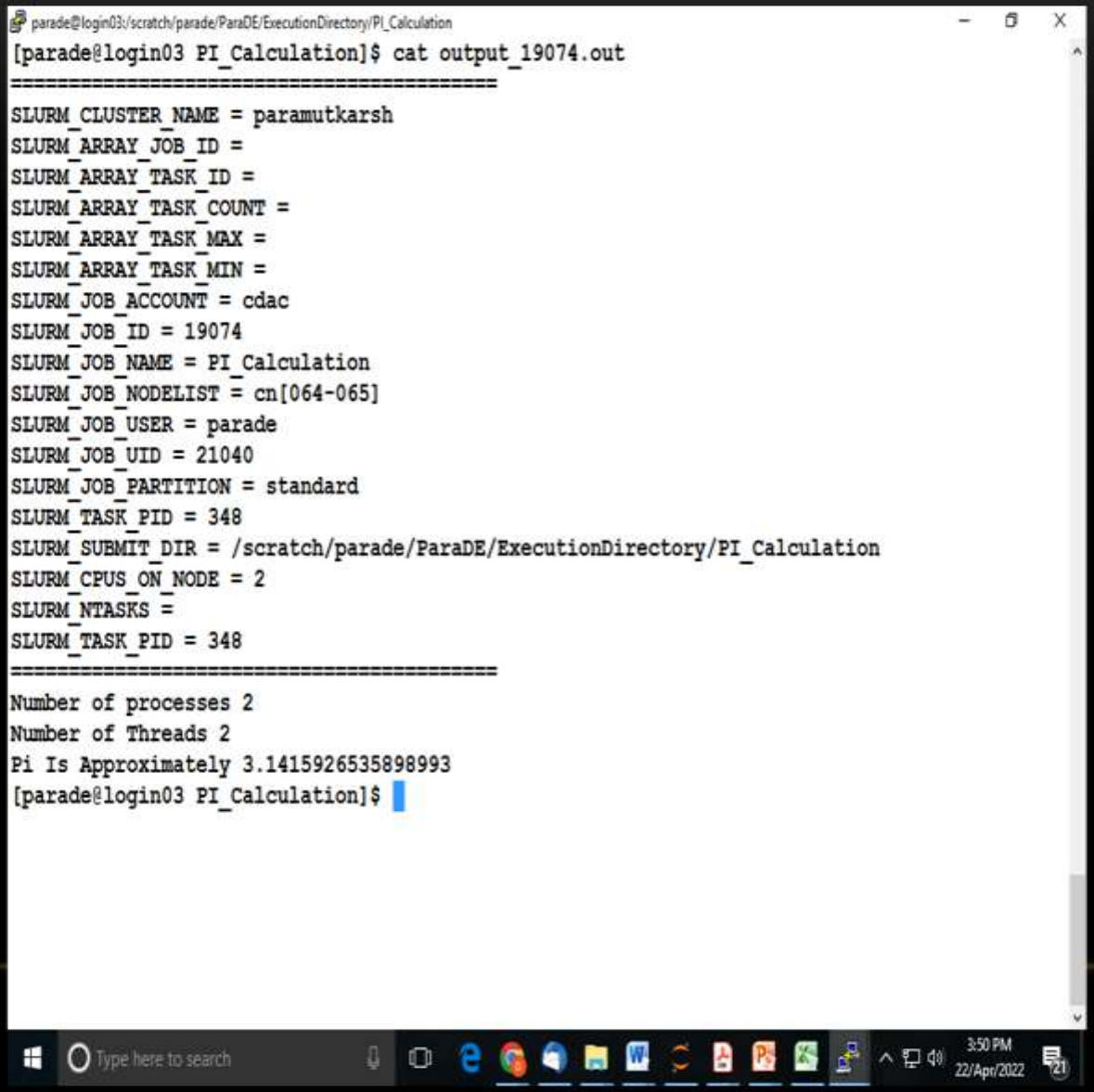
```
[parade@login03 ~]$ module avail

------------------------------ /opt/ohpc/pub/modulefiles ------------------------------
   EasyBuild/3.9.4                          oneapi/compiler32/2021.2.0
   apps/mom_6/intel_18.2                    oneapi/dal/2021.2.0
   apps/mpiblast/1.6.0/intel_18.2           oneapi/debugger/10.1.1
   apps/roms/3.6/intel_18                   oneapi/dev-utilities/2021.2.0
   autotools                                oneapi/dnnl-cpu-gomp/2021.2.0
   charliecloud/0.11                        oneapi/dnnl-cpu-iomp/2021.2.0
   charliecloud/0.22                (D)     oneapi/dnnl-cpu-tbb/2021.2.0
   clustershell/1.8.2                       oneapi/dnnl/2021.2.0
   cmake/3.15.4                             oneapi/dpct/2021.2.0
   compiler/hpc_sdk/nvhpc/21.7              oneapi/dpl/2021.2.0
   compiler/intel/2017.7.259                oneapi/init_opencl/2021.2.0
   compiler/intel/2018.4.057                oneapi/inspector/2021.2.0
   compiler/intel/2019.5.281                oneapi/intel_ipp_ia32/2021.2.0
   compiler/intel/2020.4.304        (D)     oneapi/intel_ipp_intel64/2021.2.0
   cuda/7.5                                 oneapi/intel_ippcp_ia32/2021.2.0
   cuda/8.0                                 oneapi/intel_ippcp_intel64/2021.2.0
   cuda/9.0                                 oneapi/itac/latest
   cuda/9.2                                 oneapi/itac/2021.2.0             (D)
   cuda/10.0                                oneapi/mkl/latest
   cuda/10.1                                oneapi/mkl/2021.2.0              (D)
   cuda/10.2                                oneapi/mkl32/2021.2.0
   cuda/11.2                        (D)     oneapi/mpi/2021.2.0
   gcc/8.2.0                                oneapi/oclfpga/2021.2.0
   gnu8/8.3.0                               oneapi/tbb/2021.2.0
   hwloc/2.1.0                              oneapi/tbb32/2021.2.0
   intel/18.0.5.274                         oneapi/vpl/2021.2.2
   lib/netcdf_c/4.3.3.1/intel_18            oneapi/vtune/2021.2.0
   lib/netcdf_fortran/4.4.0/intel_18        openmpi/3.1.5
   lib/parallel_hdf5/1.8.21/intel_18        openmpi/4.0.5
```

- Write code on editor or import existing code to the cluster
- Search for compatible compilers and libraries to compile
- Now compile and create executable

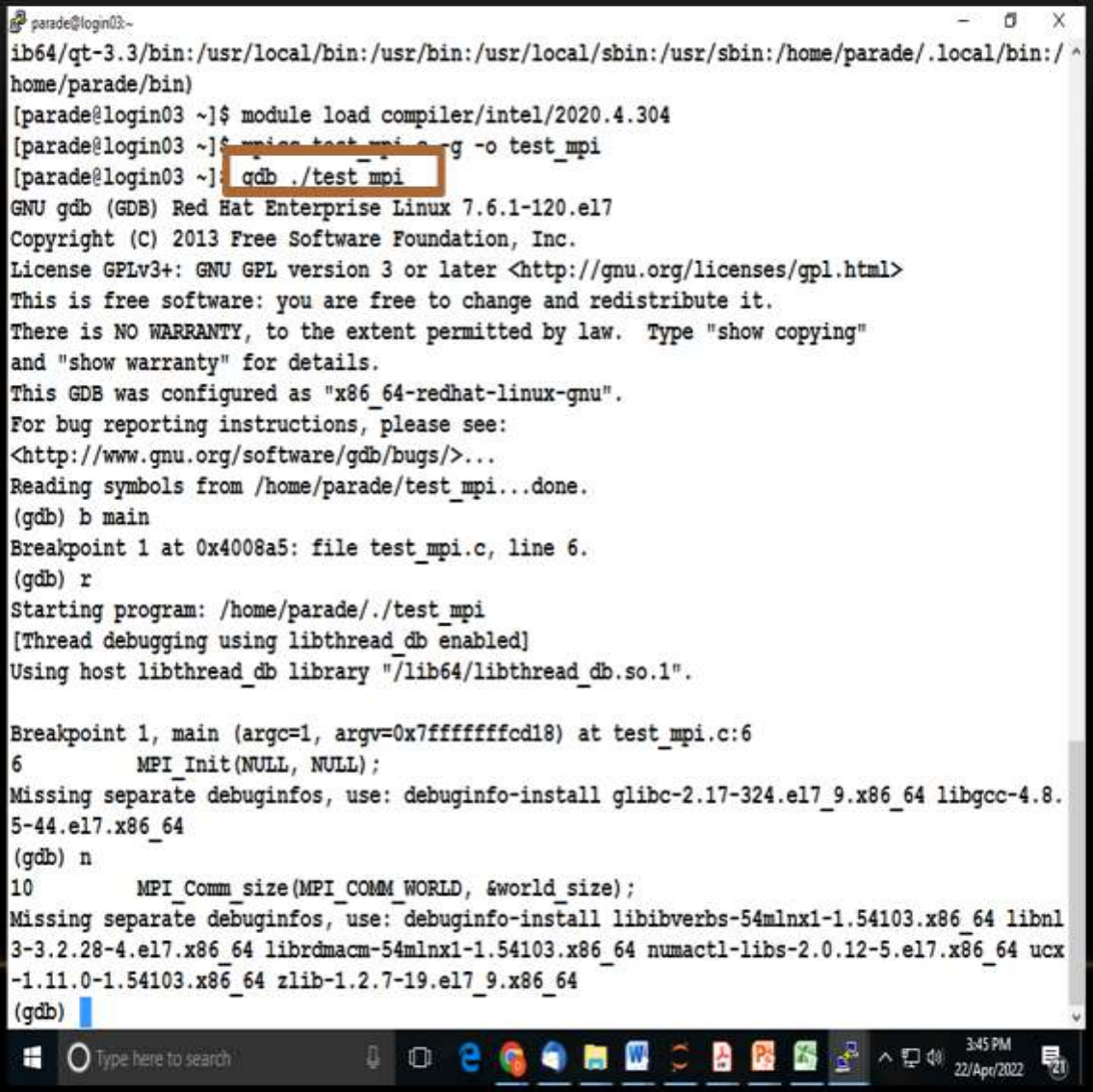
- Understand the submission process on the cluster
- Create a LRM based script to execute the application

```
parade@login03:~
[parade@login03 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
standard*    up 3-00:00:00     10 drain* cn[030-031,067-069],gpu010,hm[017,026,038-039]
standard*    up 3-00:00:00    106  alloc cn[001-028,033-060,070-072,074-101,105-110],hm[001
,025,027-037]
standard*    up 3-00:00:00     43   idle cn[029,032,061-066,073,102-104],gpu[001-009],hm[00
2-016,018-024]
gpu          up 3-00:00:00      1 drain* gpu010
gpu          up 3-00:00:00      9   idle gpu[001-009]
hm           up 3-00:00:00      4 drain* hm[017,026,038-039]
hm           up 3-00:00:00     13  alloc hm[001,025,027-037]
hm           up 3-00:00:00     22   idle hm[002-016,018-024]
cpu          up 3-00:00:00      5 drain* cn[030-031,067-069]
cpu          up 3-00:00:00     93  alloc cn[001-028,033-060,070-072,074-101,105-110]
cpu          up 3-00:00:00     12   idle cn[029,032,061-066,073,102-104]
[parade@login03 ~]$ vi slurm_script
```

View the result and optimize if required



Debug code in case of errors

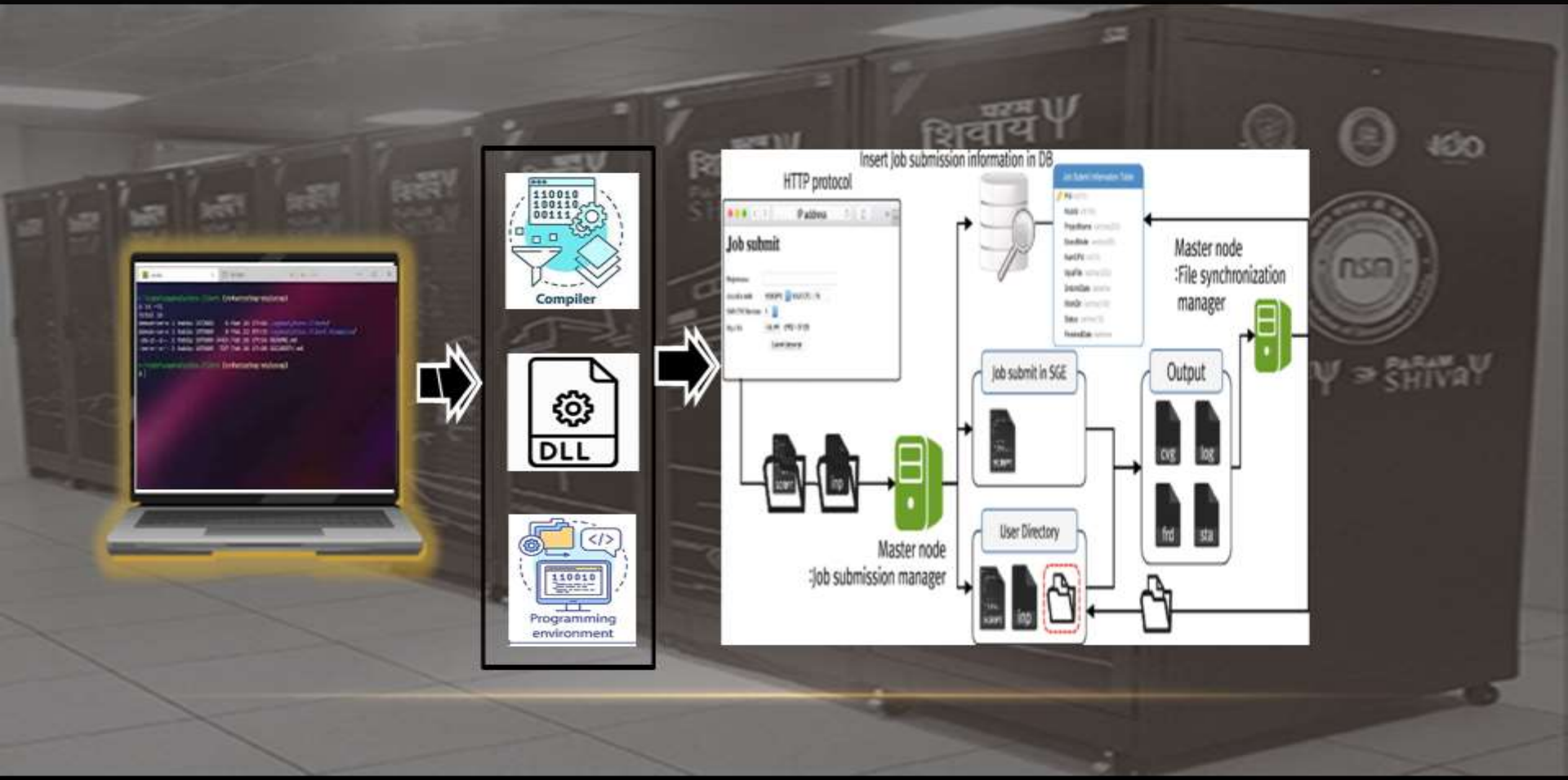


Profile the code to analyze the program for decreasing the execution time

3. paramganga.iitr.ac.in

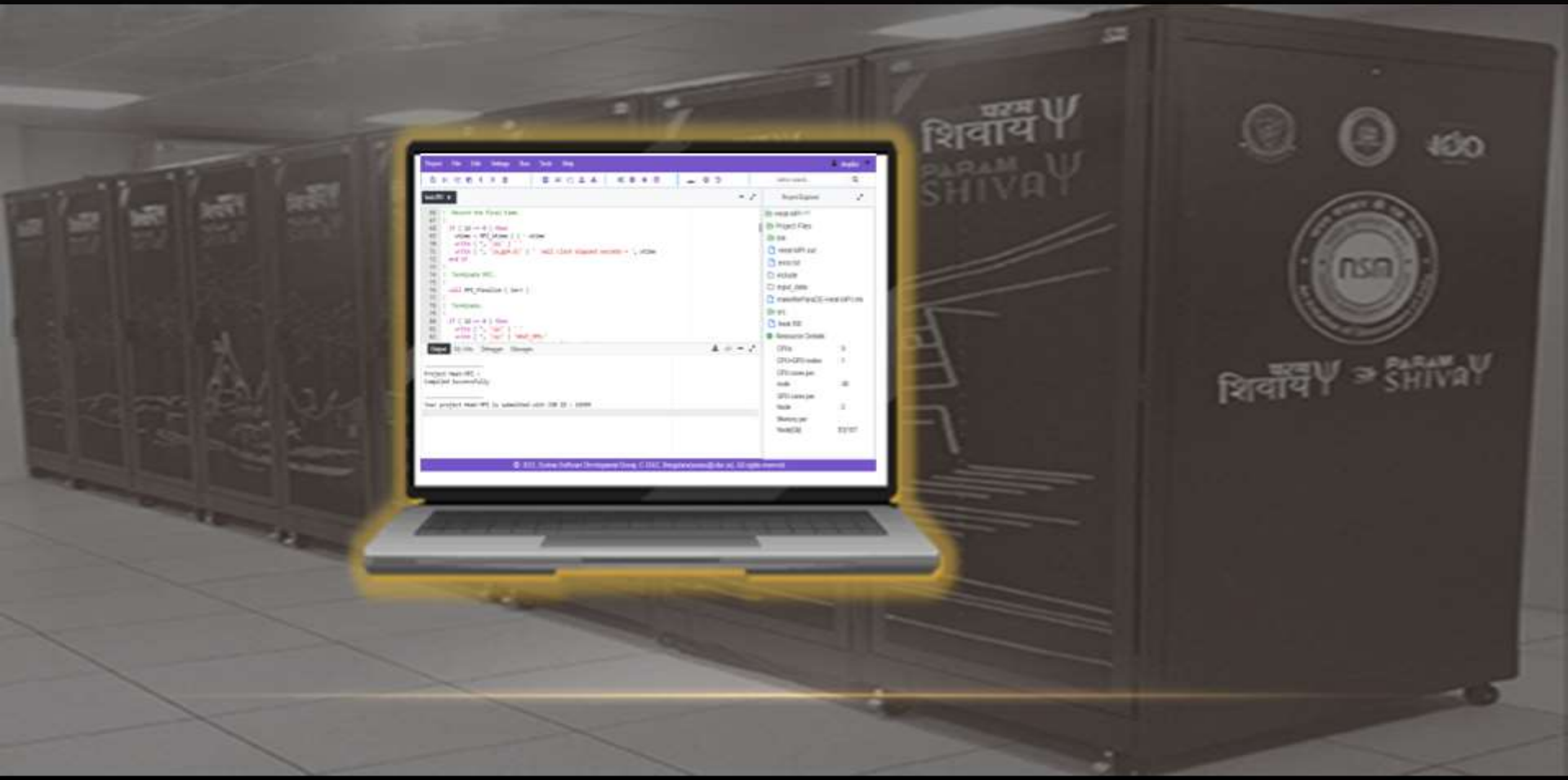
```
[parade@login04 ~]$ gcc -pg mg_serial.c -o mg_serial -lm
[parade@login04 ~]$ ./mg_serial ^C
[parade@login04 ~]$ gprof -b mg_serial gmon.out >profile.log
[parade@login04 ~]$ cat profile.log
Flat profile:

Each sample counts as 0.01 seconds.
  %   cumulative   self              self     total
 time   seconds   seconds    calls  us/call  us/call  name
100.19      0.01     0.01      101    99.20    99.20  compute
  0.00      0.01     0.00   999900     0.00     0.00  dist
  0.00      0.01     0.00      100     0.00     0.00  update
  0.00      0.01     0.00        2     0.00     0.00  cpu_time
  0.00      0.01     0.00        2     0.00     0.00  timestamp
  0.00      0.01     0.00        1     0.00     0.00  initialize
  0.00      0.01     0.00        1     0.00     0.00  r8mat_uniform_ab

                     Call graph


granularity: each sample hit covers 2 byte(s) for 99.81% of 0.01 seconds

index % time    self  children    called     name
                0.01    0.00     101/101         main [2]
[1]    100.0    0.01    0.00     101         compute [1]
                0.00    0.00  999900/999900      dist [3]
-----------------------------------------------
                                                <spontaneous>
[2]    100.0    0.00    0.01                 main [2]
                0.01    0.00     101/101         compute [1]
                0.00    0.00     100/100         update [4]
                0.00    0.00       2/2           timestamp [6]
                0.00    0.00       2/2           cpu_time [5]
                0.00    0.00       1/1           initialize [7]
-----------------------------------------------
                0.00    0.00  999900/999900      compute [1]
[3]      0.0    0.00    0.00  999900         dist [3]
-----------------------------------------------
```

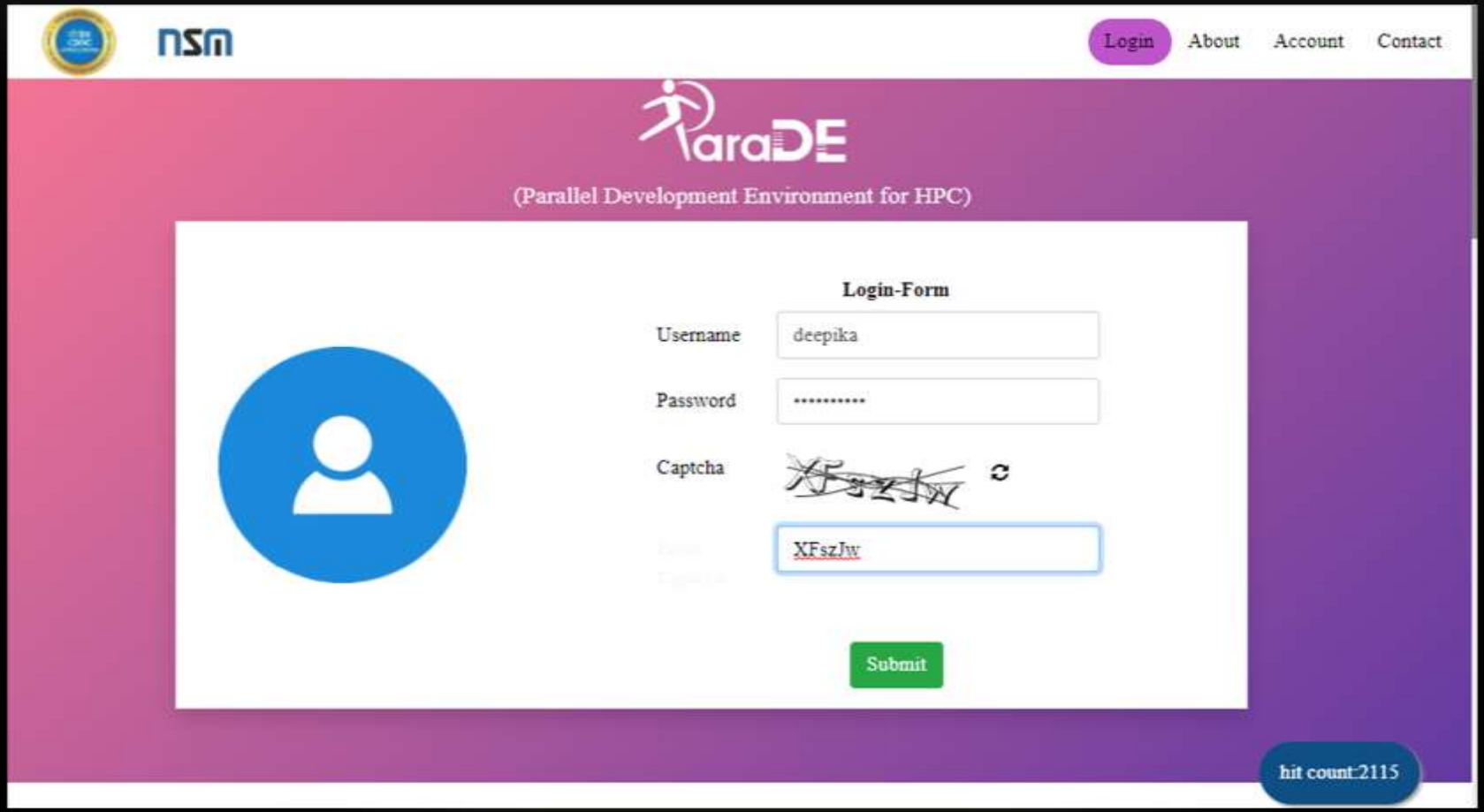
PROGRAMMING ON A SUPERCOMPUTER

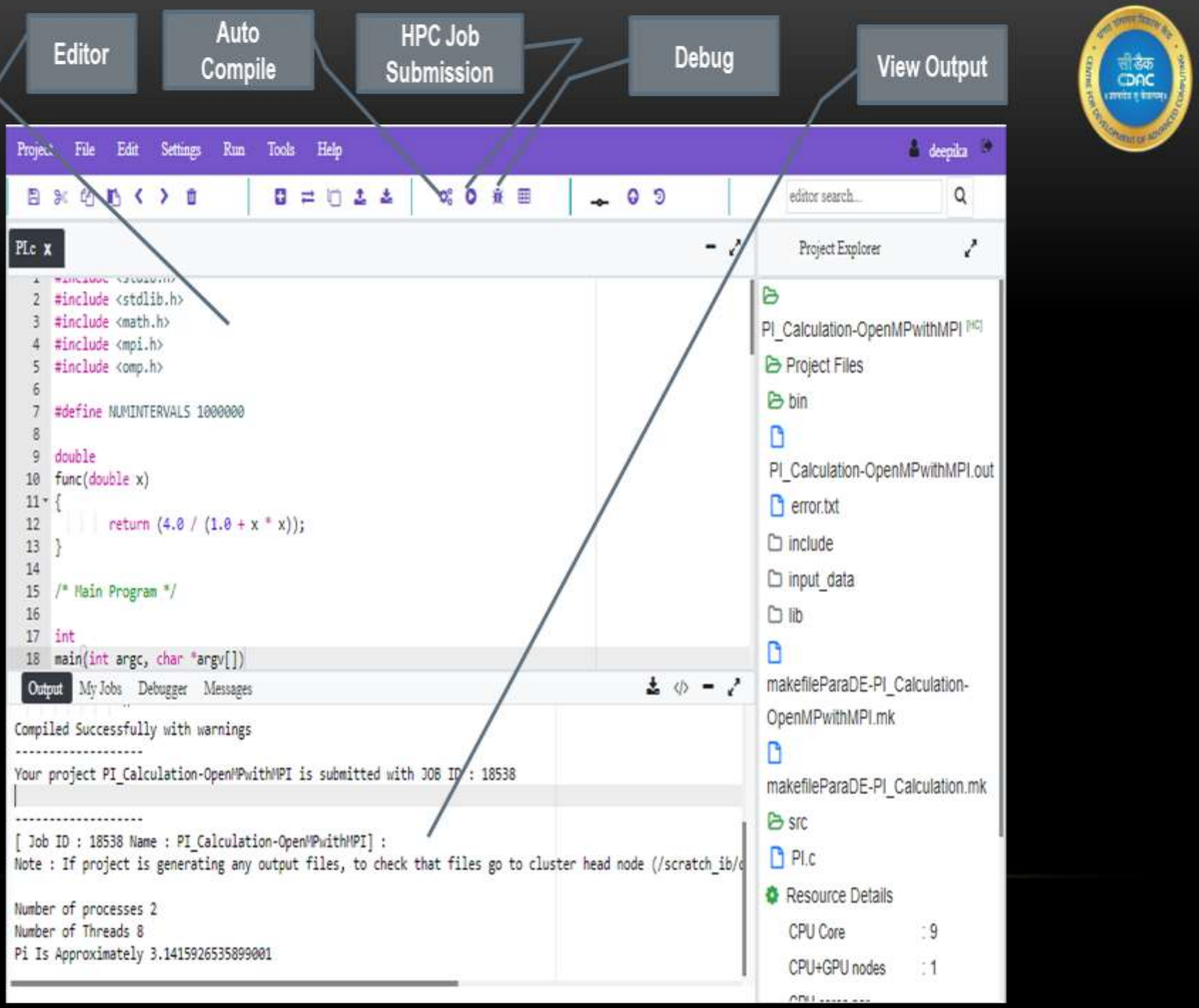


PROGRAMMING ON A SUPERCOMPUTER USING PARADE

PROGRAMMING ON A SUPERCOMPUTER USING PARADE




nsm
Login
About
Account
Contact
ParaDE
(Parallel Development Environment for HPC)
Login-Form
Username
deepika
Password
Captcha
XFszJw
Submit
hit count:2115


Editor
Auto Compile
HPC Job Submission
Debug
View Output
Project File Edit Settings Run Tools Help
deepika
editor search...
PLc x
#include <stdlib.h>
#include <math.h>
#include <mpi.h>
#include <omp.h>
#define NUMINTERVALS 1000000
double
func(double x)
{
return (4.0 / (1.0 + x * x));
}
/* Main Program */
int
main(int argc, char *argv[])
Output My Jobs Debugger Messages
Compiled Successfully with warnings
Your project PI_Calculation-OpenMPwithMPI is submitted with JOB ID : 18538
[Job ID : 18538 Name : PI_Calculation-OpenMPwithMPI] :
Note : If project is generating any output files, to check that files go to cluster head node (/scratch_ib/
Number of processes 2
Number of Threads 8
Pi Is Approximately 3.1415926535899001
Project Explorer
PI_Calculation-OpenMPwithMPI [HC]
Project Files
bin
PI_Calculation-OpenMPwithMPI.out
error.txt
include
input_data
lib
makefileParaDE-PI_Calculation-OpenMPwithMPI.mk
makefileParaDE-PI_Calculation.mk
src
PI.c
Resource Details
CPU Core : 9
CPU+GPU nodes : 1

DEMO

ParaDE URL :: https://paramuthkarsh.cdacb.in:8447/parade

IDE FOR HPC

- Anytime
- Anywhere
- Single interface for HPC programming