Subject: Parallel Programming **Date:** 12/2

① Log into another computer without password:

- ssh keygen -type rsa
  (passwordless login)
- disable firewall

- append public to authorized keys
  cat rsa public (if you have more than 3 machines) you have to append

100 - au_k pub 1, pub 2 append keys

101 - au_k pub 0, pub 2 append keys

102 - au_k pub 0, pub 1 append keys

cat >> authorized_keys (will append the keys instead of overriding the keys)

② Have MPI installed (open MPI)

openmpi and openmpi_devel ← Packages

LD_LIBRARY_PATH = $ LD_LIBRARY_PATH : /usr/lib/openmpi/lib

https://computing.llnl.gov/tutorials/mpi
lbl.gov

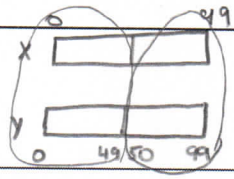
cat hosts

Subject: mpi apr in caps **Date:**

```
int main ( int argc, char *argv[])
    int num_procs, rank, namelength
    char processor_name [mpi_max_processor_name]

    mpi_init (&argc, &argv)
    mpi_comm_size (mpi_comm_world, &numprocs)
    mpi_comm_rank ( mpi_comm_world, &rank)
    mpi_get_processor_name (processor_name, &namelength)

    if (rank == 0)
        printf ("[%d/%od %os): master %od: Hello\n", rank+1, num_procs,
            processor_name, rank);

    else
        printf ("(%d /%od %os): worker %od: world!. \n", rank+1, num_procs,
            processor_name, rank);

    MPI_Finalize ()

# include <mpi.h>
# include <unistd.h>
# include <stdio.h>
```

Subject:

dot 1000



Date:

dot_product
- nprcs (# processes)
- pid (processor id)
- n (problem size)

s_idx = work * pid
work = n/nprcs
e_idx = s_idx + work

dot_product (s_idx, e_idx, x, y)

my_prod = dot_product (s_idx, e_idx, x, y)

int prod = my_prod

master _____ worker
collect

pid=0: recv

pid 1: send

for (i=1; i<nprcs; i++)
prod = prod + recv ( )

If not master:
(my_prod)
send 1 of type integer
to master.

mpi_send (my_prod, 1, int,
master, 1234, world)
to (var, # type)
↓ color? matches with recv
↓ ourselves only

If (pid == master)
for (i=1; i<nprcs; i++)
prod = prod + MPI_Recv (1, int, i, 1234, world)
# values, type, matches send, in this "network"