Introduction to parallel and distributed systems

Laboratory 1

Aim: The purpose of the lab 1 is to guide how to use computing cluster and run simple programs.

Server address (Halite) - halite.ii.pwr.edu.pl (156.17.130.39 port 22)

1. Compilation of MPI programs

Compilation of programs written in MPI is done with the command:

```
mpic++ <file_name.c> -o <name_of_executable_file>
```

An example:

```
mpic++  mpi1.cpp -o mpi1
```

2. Running MPI programs

Programs written in MPI runs on the nodes, which can be processors, cores, etc. It is a good practice when program is able to take advantage of all available nodes. To check the algorithm efficiency it should be ensure that only one MPI program use available nodes. To ensure this you need to run program in batch mode, providing specific information in the configuration file used by the system job queue. In our case we use PBS system and the configuration file should include at least the following lines (the file extension is .pbs):

```
#PBS -l nodes=<number_of_nodes>:ppn=<number_of_cores_in_node>
#PBS -N <job_name>
#PBS -j <connecting_I/O_streams>
mpiexec <path_to_the_executable program>
```

Node represents computer with processor. When you want to use only one core in each node, you need to skip in the first line the part of command from the colon to the end.
An example PBS file looks as below (let assume it is stored under `mpi1.pbs` name and used directory ~ /)

```
#PBS -l nodes=4:ppn=4
#PBS -N cnt
#PBS -j oe
mpiexec ~/mpi1
```

The `'-j oe'` means that the output stream and the error streams are combined, which will lead to the generation only one file instead two separated files.
Running on a cluster is queued to run on each node not more than one program. Of course, if there is a 16 nodes, it can, for example four programs run simultaneously, each of which uses four nodes.

To run the program using PBS is done by the following command:

```
qsub <file.pbs>
```

for example:

```
qsub mpi1.pbs
```

This inserts your job into the queue - Pbs program (and settings). As a result the job ID will be generated.

To check the programs status queue on a cluster use:

qstat –an

The `'-an'` allows you to see more information. Used by the program nodes can not be used by any other program.

To remove program from the queue use:

qdel *<process_id>*

3. An example program that you can run

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

int main(int argc, char **argv)
{
        int myrank;
        int comm_size;
        MPI_Init(&argc, &argv);
        printf("start\n");
        MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
        MPI_Comm_size(MPI_COMM_WORLD, &comm_size);
       printf("myrank = %d\n", myrank);
        MPI_Finalize();
        return 0;
}
```

I recommend you to use `pico` as an editor, and `mc` that works as Norton Commander