Università della Svizzera italiana

Institute of Computing CI

High-Performance Computing Lab **Institute of Computing**

Student: Paolo Deidda Discussed with: FULL NAME

Solution for Project 1

HPC Lab — Submission Instructions
(Please, notice that following instructions are mandatory: submissions that don't comply with, won't be considered)

- Assignments must be submitted to iCorsi (i.e. in electronic format).
- Provide source files (e.g. C/C++ files, Matlab). If you are using libraries, please add them in the file. Sources must be organized in directories called:
  Project_number_lastname_firstname
  and the file must be called:
  project_number_lastname_firstname.zip
  project_number_lastname_firstname.pdf
- The TAs will grade your project by reviewing your project write-up, and looking at the implementation you attempted, and benchmarking your code's performance.
- You are allowed to discuss all questions with anyone you like; however: (i) your submission must list anyone you discussed problems with and (ii) you must write up your submission independently.

In this project you will practice memory access optimization, performance-oriented programming, and OpenMP parallelizaton on the Rosa Cluster .

Contents

1. Rosa Warm-Up *(5 Points)*

1.1. exercise 1

The **module system** is a utility that allows the user to dynamically manage their software environment on the Rosa HPC cluster and to load different compilers, libraries and applications in order to modify environment variagles (like PATH, LD LIBRARY PATH, MANPATH, etc.) without creating conflicts between different software versions or dependencies.

As riported in the USI resource page the module system provides several commands to manage the environment.

- `module avail` – lists all available modules (on the current system)
- `module list` – lists all currently loaded modules
- `module show` – display information about
- `module load` – loads module
- `module switch` – unloads, loads
- `module rm` – unloads module
- `module purge` – unloads all loaded modules

1.2. exercise 2

The **Slurm** (Simple Linux Utility for Resource Management) is a job scheduler for Linux clusters. Main features of Slurm are:

- Job scheduling and resource management
- Framework for starting, executing, and monitoring work (jobs) on a set of allocated nodes
- Queuing management to handle multiple users and jobs

The two main components are:

- *sulurmd*: the deamnon that runs on each compute node responsible for launching, monitoring, and terminating jobs
- *slurmctld*: the central management daemon that manages job queues and allocates resources

Main commands:

- `srun`: submit a job for execution
- `sbatch`: submit a batch job
- `squeue`: view the status of jobs in the queue
- `scancel`: cancel a job
- `salloc`: allocate resources for an interactive job

1.3. exercise 3

The program is at `src/1-Rosa-warm-up/hello_world.c` and the code is the following:

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

int main(void) {
    char hostname[256];
    gethostname(hostname, sizeof(hostname));
    printf("Hello World from host: %s\n", hostname);
}
```

We can process the script with the following command:

```
srun -N1 --time=00:01:00 ./hello_worldc > hello_worldc.out 2> hello_worldc.err
```

and we can see from the output that the program has been correctly compiled and executed on the cluster on node `icsnode22` from the *slim* partion.

```
Hello World from host: icsnode22
```

1.4. exercise 4

We can see the output of the command sinfo here below:

```
PARTITION    AVAIL  TIMELIMIT   NODES  STATE NODELIST
slim*           up  2-00:00:00      7    mix icsnode[22,27-28,32-35]
slim*           up  2-00:00:00     12  alloc icsnode[17-21,23-26,29-31]
slim*           up  2-00:00:00      2   idle icsnode[36-37]
lr-slim         up  30-00:00:0      1   idle icsnode38
gpu             up  2-00:00:00      8    mix icsnode[05-06,08-13]
gpu             up  2-00:00:00      2   idle icsnode[14-15]
fat             up  2-00:00:00      4   idle icsnode[01-04]
bigMem          up  2-00:00:00      2   idle icsnode[07,15]
debug-slim      up     4:00:00      1   idle icsnode39
debug-gpu       up     4:00:00      1   idle icsnode16
multi_gpu       up  2-00:00:00      2   idle icsnode[41-42]
```

As we can see nodes are divided in partitions with names that already give us some information about their characteristics. As explained in the sbatch guide on slurm website, we can use different flags on the sbatch command to specify the partition to use with different commands.

The flag that applies to our case is:

```
sbatch --partition=fat job_script.sh
```

Submit with bigMem partition to run on nodes with very large memory:

```
sbatch --partition=bigMem job_script.sh
```

Similarly, for GPU partitions:

```
sbatch --partition=gpu job_script.sh
```

For example, we can modify the script file `slurm_job_one.sh` to specify the partition with the gpu partition with the following line:

```
#SBATCH --partition=gpu
```

1.5. exercise 5

2. Performance Characteristics *(30 Points)*

3. Optimize Square Matrix-Matrix Multiplication *(50 Points)*

4. Quality of the Report *(15 Points)*