HPC for Data Science Project – Parallel Bat Algorithm

Plan:

- First write and execute the bat algorithm as such (without HPC)
- Then do it using parallel programming models (MPI: distributed memory parallelism) way meaning MPI method (and execute via VPN to the HPC clusters machine of the university)
- (Optional): Do it with OpenMP

Info from slides:

Lecture 3b:

HPC Cluster Policy for Students (I)

1. Students must pay close attention to the level of space used in their home directory, which must not exceed 10 GB!
2. Students will be given a deadline for their access. Immediately after the expiration date, their accounts will be removed from the system with no possibility of recovery.
3. Students will have to backup elsewhere “before the deadline” their code and any data they may still need later on.
4. In any case, do not keep important data under your account. If we find problems with excessive use of disk space that can compromise the operation of the cluster, we reserve the right to lock the user and remove the account immediately. – Keep in mind that an unintentional error in the code can cause situations like those just described so don't feel exempt from this problem.
5. It is strictly forbidden to use the cluster resources for purposes other than those didactic foreseen in the HPC4DS course. Failure to follow this rule can have serious consequences. At the very least, your account will be deleted.

HPC Cluster Policy for Students (II)

6. The HPC cluster can also be accessed from outside the university network, you must connect with the VPN
7. The access to the HPC cluster and to the VPN are directly connected to the student's status.
8. Students will only be able to submit jobs on certain queues in the cluster, which differ in the maximum walltime and the number of hosts serving them. – Please use: short_cpuQ max 6h of wall time. 130 hosts. Less congested. Waiting times typically acceptable.
9. The software installed on the cluster can be found at https://sites.google.com/unitn.it/hpc/software
10. The System Team at the Data Center will NOT be able to provide support to students: unlike what is reported on the HPC cluster site, do NOT write to gestione.sistemi@unitn.it. – Any problems and/or difficulties should be reported to the instructor (sandro.fiore@unitn.it) 6

Virtual Private Network @ UniTrento

• Access to the HPC cluster @ UniTrento is provided via Virtual Private Network (VPN)
• Info about VPN@UniTrento:
• https://servicedesk.unitn.it/goauth/en?s=S00180
• https://servicedesk.unitn.it/goauth/en?kb=KB0011764
• Login node of the HPC cluster: hpc2.unitn.it

• If you try to ping it from your machine, you'll get: sfiore$ ping hpc2.unitn.it ping: cannot resolve hpc2.unitn.it: Unknown host
• which means you are not able to reach it.

1) By installing the Global Protect VPN on your machine:
2) and signing in (using UNITN Credentials)
3) you will get connected to the VPN

• Now, if you try again to ping it from your machine you'll get:

```
sfiore$ ping hpc2.unitn.it PING hpc2.unitn.it (192.168.115.242): 56 data bytes 64 bytes from 192.168.115.242: icmp_seq=0 ttl=60 time=31.208 ms 64 bytes from 192.168.115.242: icmp_seq=1 ttl=60 time=38.139 ms
```

• which means you can now reach it.
• If you got to this point, good job! You were able to successfully install the VPN on your machine.
Three key things about the login node Don't forget them!
• It is forbidden to run your software applications on the login node – This is true both for sequential and parallel job
• You must use the qsub command and run your application on the compute nodes
• Login node must be used only to edit your code, build it and run PBS CLI for the execution

Lecture 5:

Copying a file from your local machine to the cluster:

- Syntax: scp yourfile your_account@login_node:your_home_directory

- Example: scp readme.txt sandro.fiore@hpc2.unitn.it:/home/sandro.fiore

Login to the HPC cluster

- Turn on VPN
- Check with ping if the login node is reachable:
    sfiore$ ping hpc2.unitn.it
    PING hpc2.unitn.it (192.168.115.242): 56 data bytes 64 bytes from 192.168.115.242: icmp_seq=0 ttl=60 time=31.208 ms
    …
- Then run ssh using your 'username' e pwd (your UniTrento credentials):
    ssh sandro.fiore@hpc2.unitn.it

Lecture 12:

Project template (I)

- **Introduction**. An accurate description of the assigned project should be provided, including analysis of the sequential algorithm that solves the problem addressed in the project.
  - Pseudo-code, examples, graphs, figures, application instances etc. may be provided.
- **Parallel design**. Preliminary study about the opportunities for parallelism inherent in the problem sequential algorithm.
  - State of the art analysis should be performed on parallel design strategies
  - Alternative designs, related to different parallelization strategies, should be considered and discussed at this stage, appropriately motivating why some operations lend themselves to effective parallelization and others do not as well as why some data structures may or may not minimize the burden of communication and synchronization.
  - Reference to prior work as well as link with related work must be clearly stated in the report.
  - Hybrid parallelization strategies, with data dependencies must be discussed too

Project template (II)

- **Implementation**. C implementation (C++ is also allowed). The code must be properly commented. In case the student identifies multiple parallelization strategies, several implementations can be provided discussing pros and cons of each strategy. The report can include some code snippets related to the most critical and interesting parts.
  - A link to the repo must be provided too.
  - Hybrid parallelization is recommended though it is not mandatory

- **Performance and scalability analysis**.The student must analyze the performance of the developed implementation in terms of execution time, speedup, and efficiency.
  - Both strong scalability and weak scalability should be evaluated where possible.

Evaluation and significance of results

- The **evaluation** will also take into account:
  - Clarity and effectiveness of presentation;
  - Depth, correctness and originality of theoretical analysis;
  - Technical skills and documentation of implementation;
  - Number and quality of multiple parallel strategies, if any;
  - Critical thinking in performance evaluation and analysis;
- **Significance of results**: since parallel programming is primarily concerned with performance, a good project must also provide a significant speedup.
- Note: The report should be as concise as possible, but without detriment to clarity of presentation. Expected size: 10 pages.

Relevant aspects for your project

- Cite references regarding papers or other material you got inspired by (state of the art analysis is part of the report!)
- Cite data sources you are using for your project
- Finding input data is challenging? Create a simple random generator app to create synthetic data
  - Different datasets (small, medium, large size) and then compare the performance of your application on the 3 input data
- Replicating/inspired by an existing work? Some tips
  - Reproduce and compare the results. Comment differences and other findings
  - Start from pseudocode and provide your C implementation; compare with findings from the authors
  - Add from existing code and add your own delta; then compare
  - Take 2 different implementations and port them on the cluster; compare the two on the cluster + original work

Project and oral examination

- *Project*:
  - parallel application using MPI (and OpenMP)
    - OpenMP implementation is not mandatory (bonus)
  - to be developed in team (1-2 students)
  - will be discussed during the oral examination
  - All the team members must be present during the oral examination and the project presentation