Parallel/GPU Computing Lab

Many of you likely already have an idea of what you want to use HPC for. If you already know the software you intend to use, that is great. However, if you have a broad idea, this lab could be even more beneficial as you can investigate different tools for your needs. The aim of this lab is to help you better understand the software that you are going to use for your research. The software you choose needs to be parallelizable or executable on GPUs! If you're having a hard time choosing software to analyze, there is a list of well-known HPC software at the bottom of the page that you could pick from.

It should be possible to answer all of these questions using the documentation of the software. Many computing centers provide additional information on how to use different software packages. Googling "<Software name> Slurm" often finds these pages, which can be decent guidelines for initial configurations.

Do not answer the question with just "yes" ("no" is sometimes applicable)! Fill in the

Feel free to ask questions in Slack if you get stuck, especially on the execution part.

Software Analysis Questions

1. **Username on Rocket:**
2. **Name of software:**
3. **Repository/Distribution website (GitHub, homepage, etc.):**
4. **Website for documentation:**
5. **Planned usage with CPU or GPU:**
6. **Is the software available as a module on Rocket (answer with module name)?**
7. **Installation Methods:**
    a. How can the software be installed (e.g., CMake, Make, pip, conda, apt)?
    b. Are there any specific dependencies or prerequisites?
    c. Can you find a container that provides the software?
8. **Programming language(s) used in the implementation:**
9. **Computational Capabilities:**
    a. Can the software use threads (e.g., OpenMP, pthreads)?
    b. Can the software use multiple processes for computation (e.g., MPI, multiprocessing in Python)?
    c. Is the software capable of distributed computation (across multiple nodes)?
    d. Can the software use GPUs?
    e. Does the software support using multiple GPUs? If so, does it support multiple GPUs per process, or 1 GPU per process and communication between them.
    f. Is distributed (multi-node) computing (with GPUs) available?
10. **Communication Methods:**
    a. What method of communication is used for distributed computation (e.g., MPI, shared memory, RPC)?
    b. What hardware can the communication use (e.g., Infiniband, TCP, RDMA)?
11. **Scalability:**
    a. Is there any documentation, study or benchmark data available on scalability?
    b. If yes, how does the software scale with an increasing number of cores or nodes? Is it better for GPUs?
12. **Hardware and Platform Support:**
    a. What types of hardware architectures does the software support (e.g., x86, ARM, GPUs)?
    b. Is it available for Linux or Windows?

c. If the software supports GPU acceleration, can it use AMD and NVIDIA GPUs?

13. **Performance Tuning:**
    a. Are there any performance tuning parameters or settings available (e.g., environment variables, compiler flags)?
    b. Is there a performance profiling tool integrated or recommended for use with the software?
14. **Licensing and Availability:**
    a. Does the software require a license to use?
    b. Are there any usage restrictions, especially in a commercial or academic context?
15. **Execution:**
    a. Are there any tutorials, example scripts, or user communities available (insert link)?
    b. Try to execute the software using threading, multiprocessing or GPUs! Ideally, try to find a sample dataset or synthesize your own, so that the run takes some time. This way you can analyze the effect different parameters have on the performance. Make a batch script that will run it and fill in the following blanks.
        i. Path to batch script:
        ii. JobID of successful run:
        iii. Path to output of the run:
        iv. Path to data:
16. **Measuring scaling (optional):**
    a. Try out different launch parameters to see how they perform. Map out the results and try to find the “sweet spot”.
17. **Feedback:**

List of Well-Known HPC Software for Analysis:

Recommended (we will be able to assist you more): *mpi4py*, *PyTorch*, *TensorFlow*

More difficult (will likely require quite a bit more work from your side)*: GROMACS*, *OpenFOAM*, *LAMMPS*, *MATLAB*, *VASP*, *parallel processing packages for R (or any language)*