This covers the basic concepts of High Process Computing Systems and how the Hellgate Cluster is laid-out..

**HPC Carpentry (Resource)**

Great information about using HPC systems.

[HPC-Carpentry.org](https://www.hpc-carpentry.org/)

**UNIX Tutorial (Highly Recommended)**

Fantastic tutorial about using UNIX Shells.

[HPC Carpentry - Software Carpentry Unix Shell](https://swcarpentry.github.io/shell-novice)

**HPC Introduction (Optional but recommended)**

Great insight into how a HPC works and how to submit Slurm jobs, similar the Hellgate.

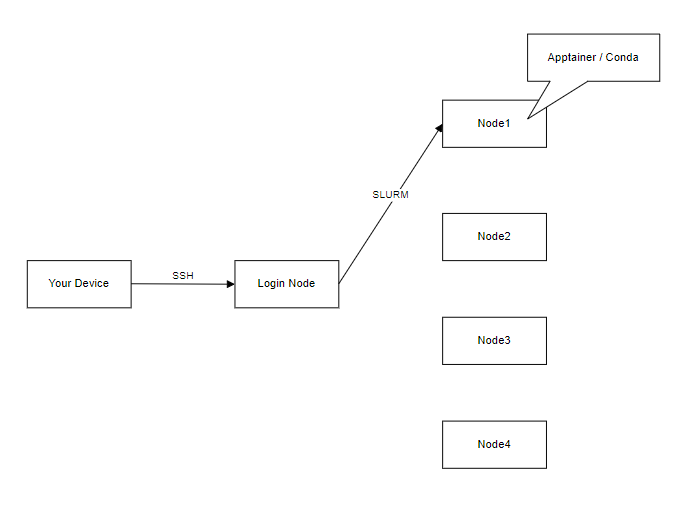
[HPC Carpentry - Introduction to High Performance Computing](https://carpentries-incubator.github.io/hpc-intro/)

**HPC Process**

The primary purpose of utilizing High-Performance Computing (HPC) systems like Hellgate is to process data faster by leveraging increased computational resources. Several components on Hellgate enable users to achieve this efficiently.

First, you start by connecting to the login node, which acts as the front door of the HPC system. Using SSH (Secure Shell), you securely access this node from your local computer. The login node is where you prepare your work, such as writing and submitting job scripts, but it’s not where the heavy computations occur.

Once on the login node, you use SLURM (Simple Linux Utility for Resource Management) to request resources from the compute nodes. These nodes are the powerful machines where your actual computations run. This is typically done with srun or sbatch.



You will quickly realize that although you have the resources to process data, you do not have the necessary software installed. Since the HPC is a shared resource, you cannot install programs directly. Different applications often require specific versions of libraries and tools, and environments allow you to manage these dependencies without conflicts. For instance, one project might need Python 3.8 while another requires Python 3.9. To address this, we use tools like Apptainer and Conda, enabling us to switch between different versions easily and ensuring that your projects run smoothly without interfering with others.

**Nodes**

In high-performance computing (HPC) systems, both CPUs (Central Processing Units) and GPUs (Graphics Processing Units) play crucial roles.

**CPU**

Typically, CPUs have a smaller number of powerful cores (usually between 2 and 64).

CPUs excel at tasks that require sequential processing and are good at handling complex, single-threaded operations.

CPU: Think of the CPU as a super-smart builder who can do many different tasks, but usually one at a time. This builder is really good at following instructions and can switch between tasks quickly. So, if we ask the CPU to build a tower, then a wall, and then a door, it can do all of these one after the other very well.

**GPU**

GPUs have thousands of smaller, less powerful cores designed for handling multiple tasks simultaneously.

GPUs are ideal for tasks that can be broken down into smaller, parallel operations, such as matrix multiplications used in machine learning and simulations.

GPU: Imagine we have a whole team of builders who are really good at doing the same task over and over again. If you need to build many identical towers at the same time, the GPU team can do it much faster because they can all work on different towers at once.

**Hellgate Nodes**

CPUs: Partition Name: ‘CPU(all)’

| **Name** | **CPUs** | **Memory (RAM)** |
| --- | --- | --- |
| hgcpu1-[1-8] | 72 | 750GB |
| hgcpu2-[1-4] | 32 | 250GB |
| hgcpu3-[1-8] | 36 | 185GB |
| hgcpu3-[9-12] | 72 | 185GB |
| hgcpu6-[1-13, 15-16] | 48 | 185GB |
| hgcpu6-14 | 96 | 185GB |
| hgcpu7-1 | 128 | 1826GB |
| hgcpu8-[1-4] | 56 | 250GB |

GPUs (NVIDIA): Partition Name: ‘GPU(all)’

| **Name** | **CPUs** | **Memory** | **CUDA Core** | **Tensor Cores** | **VRAM** | **Device** |
| --- | --- | --- | --- | --- | --- | --- |
| hggpu4-[1-12] | 36 | 500GB | 4352 | 544 | 11GB DDR6 | NVIDIA GeForce RTX 2080 Ti |
| hggpu5-1 | 48 | 500GB | 10752 | 336 | 48GB DDR6 | NVIDIA A40 |