**HPC Setup README**

**San Jose State University, Department of Computer Engineering**

**Author**: [Aishwarya Paruchuri](mailto:aishwaryaparuchuri@sjsu.edu) [Krupaben Kothadia](mailto:krupaben.kothadia@sjsu.edu)

**Project Advisor**: Dr. [Harry Li](mailto:hua.li@sjsu.edu)

| 05/08/2023 | Created Document | [Aishwarya Paruchuri](mailto:aishwaryaparuchuri@sjsu.edu) [Krupaben Kothadia](mailto:krupaben.kothadia@sjsu.edu) |
| --- | --- | --- |

SJSU’s HPC Source Link: <https://www.sjsu.edu/cmpe/resources/hpc.php>

Table of Contents

| Index | Description | Page No. |
| --- | --- | --- |
| I. | Introduction | 2 |
| II. | Accessing the HPC | 3 |
| III. | File Transfer Techniques | 5 |
| IV. | Preliminaries for a Training Session | 5 |

1. Introduction

High Performance Computing (HPC) can bring significant benefits to the training of deep learning models, but it also presents a number of challenges. Here are some of the benefits and challenges of using HPC for deep learning training:

Benefits:

1. Reduced training time: HPC enables parallel processing, which can significantly reduce the time required for training deep learning models. This can be particularly important for large-scale projects with enormous amounts of data.

2. Ability to handle large datasets: With HPC, deep learning models can be trained on much larger datasets than would be possible with traditional computing resources. This can help improve model accuracy and enable the creation of more sophisticated models.

3. Improved scalability: HPC can scale to support multiple GPUs or even clusters of GPUs, enabling deep learning models to be trained on larger datasets and at a faster pace.

4. Increased experimentation: HPC can allow researchers to experiment with a wider range of hyperparameters and model architectures, leading to better understanding of deep learning models and potentially better performing models.

Challenges:

1. High cost: HPC can be very expensive to set up and maintain, with specialized hardware and software required to achieve high performance. This can limit access to HPC for many researchers and organizations.

2. Complexity: HPC systems can be complex and require specialized knowledge to set up and maintain. This can be a barrier to entry for many researchers and organizations.

3. Limited availability: HPC systems are often in high demand, with limited availability for researchers and organizations. This can make it difficult to access HPC resources when they are needed.

4. Data transfer: Moving large datasets to and from HPC systems can be a challenge due to the size of the data and the limited bandwidth of networks. This can result in long transfer times, which can impact training time.

Overall, while HPC can bring significant benefits to the training of deep learning models, it is important to carefully consider the costs and challenges associated with using HPC before deciding to invest in these resources.

Recognizing the immense computational demands of such tasks, we leveraged SJSU's HPC resources to train and test our models, ensuring optimal performance and efficiency.

SJSU’s HPC resources are composed of:

* A total of 36 nodes, 15 include 1 NVIDIA Tesla P100 12 GB GPUs, and 1 has 2 NVIDIA Tesla P100 GPUs
* 20 nodes (compute nodes) have 128 GB of RAM, and 16 nodes (GPU and condo nodes) feature 256 GB.

1. Accessing the HPC

# Note: My computer specifications are as follows:

Operating System

* MacOS Monterey 12.6
* Chip: Apple M1
* Memory: 8 GB

1. Connect to VPN by following this guide: <https://sjsu.edu/it/docs/connectivity/How%20to%20Connect%20to%20VPN%20Using%20Cisco%20AnyConnect-For%20Macs-Students.pdf>
2. Once the connection is done, open that and select Student option and then login using your SJSU credential. Confirm that your SJSU ID has been added to the HPC access list. If not, contact your instructor or project advisor to request access.
3. Open terminal and establish SSH Connection using this command:

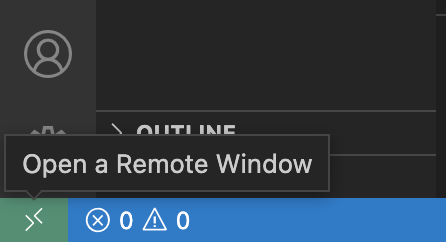
ssh SJSUID@coe-hpc1.sjsu.edu (replace "SJSUID" with your SJSU ID). Your password is the same as your SJSU account password.

1. Connect to Visual Studio Code (VS Code):

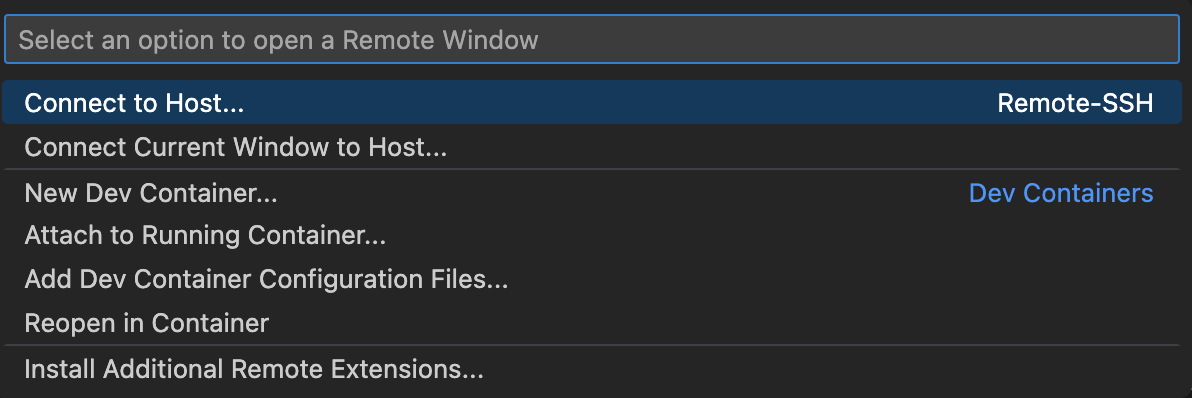
4.1. VS Code Setup: Install VS Code on your machine if you haven't already. Use the instructions provided from Microsoft found here: <https://code.visualstudio.com/download>

4.2. Extension Installation: Within VS Code, install the Remote-SSH extension. This extension is provided by Microsoft and can be found here (<https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-ssh>) or within the Extensions sidebar option within VS Code

4.3. SSH Connection Setup: Add a new SSH connection by clicking the green box in the bottom left corner of VS Code, then select “Connect to Host…” and enter “coe-hpc1.sjsu.edu”.



4.4. Select “Connect to Host…”



4.5. Enter the domain of SJSU’s HPC1, it is “coe-hpc1.sjsu.edu”.

4.6. Authentication: System will prompt for username (SJSU ID) and password (SJSU account password) upon accessing HPC.

4.7. Successful Connection: Post-authentication, view folders and working directories within the HPC on the left-hand side of VS Code.

III. File Transfer to HPC

Now that the necessary files are on your machine, you must transfer them to SJSU’s HPC.

File Transfer via FileZilla.

* 1. Once you are connected to the HPC, download filezilla and establish a connection there using your HPC connection details and use port number as 22. Add files from local to HPC and vice versa.
  2. Once this is complete, continue on with training a model. Instructions for this are found in our YOLACTEdge README here: https://github.com/adkap2/Instance-Segmentation-with-Unity-Interactive-Augmented-Reality/blob/Krupa/YolactEdge\_Instance\_Segmentation\_README.md

IV. Preliminaries for a Training Session

Once connected to the HPC and the training process is ready to start, there are a few crucial steps that need to be completed first.

1. Once connected clone the project repo

git clone https://github.com/adkap2/Instance-Segmentation-with-Unity-Interactive-Augmented-Reality.git and add SyntheticHomes data using Cyberduck or “Drag and Drop” (as per the instructions above).

1. Connect to a gpu instance:

| srun -p gpu --gres=gpu --pty /bin/bash |
| --- |

or

| srun -p gpu --time=47:55:00 --gres=gpu --pty /bin/bash |
| --- |

1. User must load python in order to run project code, this will also import the most common dependencies / libraries used within python project. Run: module load python3
2. Additionally, if the HPC does not have the dependency needed to run a program use the following to install a dependency:

| python3 -m pip install <packge\_name> |
| --- |

**Citations**

1. [1] “College of Engineering HPC,” College of Engineering HPC | Computer Engineering, https://www.sjsu.edu/cmpe/resources/hpc.php (accessed May 8, 2023).
2. [2] Microsoft, “Visual studio code - code editing. redefined,” RSS, https://code.visualstudio.com/ (accessed May 8, 2023).
3. [3] iterate GmbH, “Cyberduck: Libre server and cloud storage browser for mac and windows with support for FTP, SFTP,” Browse with Cyberduck on macOS and Windows, https://cyberduck.io/ (accessed May 8, 2023).
4. [4] adkap2, “Interactive augmented reality for semantic segmentation on indoor environments,” GitHub, https://github.com/adkap2/Instance-Segmentation-with-Unity-Interactive-Augmented-Reality (accessed May 8, 2023).