Summer of Reproducibility

MPI Appliance for HPC Research on Chameleon

Mentored By Ken Raffenetti

- Rohan Babbar
- LinkedIn - Rohan Babbar
- GitHub - rohanbabbar04
- E-Mail - rohanbabbar0408@gmail.com
- New Delhi, India (GMT + 5:30)

About me

I am a 2024 graduate with a Bachelor's degree in Information Technology and Mathematical Innovations from the University of Delhi, India. I am currently working as a Data Analyst at Stryker Corporation. I enjoy working on new ideas and projects and communicating with people around the world. I have a strong command of Git and experience in writing clean, well-documented code.

Abstract

Message Passing Interface (MPI) is the backbone of high-performance computing (HPC), enabling efficient scaling across thousands of processing cores. However, reproducing MPI-based experiments remains challenging due to dependencies on specific library versions, network configurations, and multi-node setups. To address this, we extend the work initiated for SC24's reproducibility initiative by developing standardized MPI environments for the Chameleon testbed.

The main outcomes of this project are:

- **Ready-to-Use MPI Disk Images**: Create one or more pre-configured images with the correct versions of MPI and dependencies, ensuring a consistent environment.
- **Simple Cluster Configuration Scripts**: Provide scripts or playbooks to efficiently set up a fully functional MPI cluster on Chameleon, eliminating manual setup steps.
- **Orchestration Template**: Develop an automated workflow that configures networks, instances, and additional resources needed to run large-scale MPI workloads.

Technical Details

The important technologies used in this project are listed below:

1. Chameleon - An openstack-based testbed providing bare-metal access where the MPI appliance will be hosted.
2. Openstack - For managing compute, storage, and networking resources.
3. Message Passing Interface (MPI) - A standard for distributed computing, with implementations like:
    - MPICH
    - Open MPI
    - Intel-MPI
4. Jupyter Interface - Use the jupyter interface in Chameleon to create jupyter notebooks for Experiment and Analysis.
5. Ansible - YAML based playbooks for orchestration.

6. Bash Scripts - For system initialization, setting up SSH keys, and configuring MPI
7. Python - Create Examples using python-chi.

Benefits to the community

- Pre-configured MPI environments make it easy for researchers to set up and scale distributed computing clusters.
- Shared images and templates in Chameleon and Trovi enable seamless experiment replication.

Project Implementation

1. **Create Disk Images with MPI libraries and required dependencies**
    - Start with an existing image and configure it with MPI libraries, network settings, GCC, and other necessary dependencies.
    - Create a snapshot of the instance using `cc-snapshot` once the setup is complete.
    - Test the image to ensure all packages are up to date and compatible with Chameleon's bare metal and VM environments.
    - Publish the finalized image in the Appliance Catalog and publish it in Trovi for reproducible experiments.
2. **Cluster Setup Scripts**
    - Create Ansible playbooks to that join new instances into an MPI Cluster
    - Configure hostnames, set up SSH keys access, and adjust MPI runtime settings.
    - Perform both simple and complex tests to validate the cluster's functionality.
3. **Orchestration Templates**

    The orchestration template automates and parameterizes the deployment of the MPI cluster in Chameleon:
    - Define network setup, instance counts, and MPI environment variables.
    - Enable parameterization for cluster size, disk images, and other variables to allow user customization.
    - Test the configuration to ensure correct provisioning of resources and MPI functionality.

- Publish the orchestration template for easy access.

4. **Documentation & Testing**
    - Use the Jupyter interface provided by Chameleon to document examples using python-chi.
    - Create an Experiment and Performance Analysis Notebook to demonstrate how workload execution varies with cluster size.
    - Test the orchestration template with multiple cluster sizes and workloads, ranging from simple "Hello World" MPI runs to complex computational tasks.

List of Deliverables

1. Disk image with MPI libraries and dependencies, published in Chameleon Appliance Catalog and Trovi.

2. Ansible playbooks for MPI cluster setup and configuration.

3. An orchestration template for automated cluster deployment.

4. Jupyter notebooks for documentation/visualization and performance analysis.

Project Timeline

Dates	Task
May 8	Community Bonding Period Starts
May 8 - June 1	• Get in touch with the assigned mentors. • Discuss and clear the remaining doubts regarding the project. • Break down the goals into smaller tasks for a more thorough analysis of the goals and the planned workflow.
June 1	Community Bonding Period Ends
June 2 - June 22	• Configure an existing image with MPI libraries, network settings, and dependencies. • Create a snapshot and test compatibility with Chameleon's bare metal and VM environments.

	• Publish the image in the Appliance Catalog and publish it in Trovi.
June 23 - July 13	• Create Ansible playbooks to automate instance joining in an MPI cluster. • Configure hostnames, SSH key access, and MPI runtime settings. • Validate cluster functionality with simple and complex tests.
July 14 - July 18	Mid-Term Evaluation
July 19 - August 10	• Work with the Chameleon team to utilize orchestration best practices for automating MPI cluster deployment in Chameleon. • Enable parameterization for network, instance count, disk images, and MPI settings.
August 11 - August 25	• Document examples • Test with multiple cluster sizes and workloads. • Finish up any pending code/tests/documentation
August 26 - September 1	Final Evaluation

Why I chose this project

I enjoy working on new ideas and have a keen interest in MPI. I have worked on a project using mpi4py before, which helped me understand distributed computing better. I have also been contributing to open-source projects for a few years. This experience will help me complete this project successfully.

Why am I a good fit for this project

1. Open Source Contributions

    1. PyLops-MPI

    As part of Google Summer of Code 2023, I contributed to integrating an MPI backend into PyLops, leading to the release of PyLops-MPI, a new library for distributed computing with PyLops. My contributions are documented in this report.

    2. PreliZ

    As part of Google Summer of Code 2024, I contributed to integrating an MPI backend into PyLops, leading to the release of PyLops-MPI, a new library for distributed computing with PyLops. My contributions are documented in this report.

2. Work Experience

    I am a Data Analyst at Stryker Corporation, where I develop core solutions for international teams, automate processes to enhance efficiency, analyze complex datasets, and create visualizations. Previously, I worked as a Python Developer at Medvolt AI, where I developed core solutions and modules for drug discovery.

Availability

I will be available throughout the SOR summer period and am willing to devote 25-30 hours per week to achieving my goals. If I fall behind my proposed timeline, I will work extra hours over the following days to get back on track. I will always be available via Slack or email

Am I applying to any other project?

No, I will not be applying for any other project in the Summer of Reproducibility 2025.

After SOR

After the SOR coding period ends, I plan to continue contributing by adding new features and fixing bugs. I also wish to help maintain the project in the future and support new contributors as they get started.