UTM

UNIVERSITI TEKNOLOGI MALAYSIA

SESSION 20242025/2 SECR3223-03

(High Performance & Parallel Computing)

Assignment 2:

Installation of C/C++ Compiler and OpenMPI Library in a Virtual Machine

Lecturer: Dr. Johan Bin Mohamad Sharif

NO	NAME	MATRIC NO.
1	MUHAMAD FAIZ BIN ABDUL MUTALIB	A21EC0059

Contents

Abstract

This report details the process of installing C/C++ compiler and OpenMPI library in a Linux virtual machine. The purpose is to establish a functional high-performance computing environment for parallel programming. The report covers the step-by-step installation process from setting up a virtual machine to verifying the successful installation of the OpenMPI library through sample program compilation and execution. This environment is essential for developing and running parallel programs in distributed computing environments.

Keywords

C/C++ Compiler, OpenMPI, Virtual Machine, Parallel Programming, High-Performance Computing, Ubuntu Linux

1 Introduction

High-performance computing and parallel programming are becoming increasingly important in the field of computer science and engineering. Message Passing Interface (MPI) is a standardized and portable message-passing system designed to function on parallel computing architectures. OpenMPI is an open-source implementation of the MPI standard that enables parallel computing across distributed environments.

This report provides a comprehensive guide to setting up a development environment for MPI programming using a Linux virtual machine. The virtual machine approach offers several advantages, including isolation from the host system, reproducibility, and the ability to experiment without affecting the main operating system. The setup process involves installing a virtual machine software, creating a virtual machine instance, installing Ubuntu Linux, configuring a C/C++ compiler, and finally installing and testing the OpenMPI library.

2 Related Content: Installation Process

2.1 Setting Up the Virtual Environment

The first step in creating a suitable environment for OpenMPI development is to establish a virtual machine:

1. **Install Virtualization Software:** Install either VirtualBox or VMWare on your host system. These applications provide platform virtualization solutions that allow running multiple operating system instances on a single physical machine.

2. **Create a Virtual Machine (VM):** Using the virtualization software, create a new virtual machine with appropriate specifications. Recommended specifications include:
    - At least 2GB of RAM allocated to the VM
    - Minimum 20GB of virtual disk space
    - 2 or more virtual CPU cores for better parallel processing performance
3. **Install Ubuntu Linux:** Download an Ubuntu ISO file (latest LTS version recommended) and install it on the virtual machine. Follow the on-screen instructions during the installation process to set up user accounts and basic system configurations.

2.2 Installing and Verifying C/C++ Compiler

Before installing OpenMPI, a C/C++ compiler must be properly installed and verified:

1. **Verify Existing Compiler:** Open a terminal window and check if a compiler is already installed using the command:

```
$ gcc -v
```

2. **Install Compiler (if needed):** If the compiler is not already installed, install the necessary packages using the following commands:

```
$ sudo apt update
$ sudo apt install build-essential
```

   Additional information on compiler installation can be found at: https://help.ubuntu.com/community/InstallingCompilers

3. **Test the Compiler:** Create a simple C program to verify proper installation:

```c
#include <stdio.h>
int main()
{
    // printf() displays the string inside quotation
    printf("Hello, World!");
    return 0;
}
```

   Save this code to a file named `helloworld.c`, then compile and run it using:

```
$ gcc -o helloworld helloworld.c
$ ./helloworld
```

The output should display: `Hello, World!`

2.3 Installing OpenMPI Library

With a functioning C/C++ compiler in place, proceed to install the OpenMPI library:

1. **Install via Package Manager:** The simplest method is to use Ubuntu's package manager:

```
$ sudo apt update
$ sudo apt install openmpi-bin libopenmpi-dev
```

2. **Alternative Installation Method:** For a custom installation, follow the detailed guide at: http://lsi.ugr.es/jmantas/pdp/ayuda/datos/instalaciones/Install_OpenMPI_en.pdf

3. **Verify OpenMPI Installation:** Check that the MPI compiler wrapper is properly installed:

```
$ mpicc -v
```

This should display version information about the MPI compiler wrapper.

2.4 Testing OpenMPI with a Sample Program

To ensure that OpenMPI is functioning correctly, create and run a simple MPI program:

1. **Create an MPI Test Program:** Save the following code as `helloworld_mpi.c`:

```c
#include <stdio.h>
#include <mpi.h>

int main(int argc, char *argv[]) {
    int mpierror, npes, myrank; /* global variables */

    mpierror = MPI_Init(&argc, &argv);
    mpierror = MPI_Comm_size(MPI_COMM_WORLD, &npes);
```

```
    mpierror = MPI_Comm_rank(MPI_COMM_WORLD, &myrank);

    printf("From process %d out of %d, Hello World!!\n",
           myrank, npes); /* global instruction */

    mpierror = MPI_Finalize();
    return 0;
}
```

2. **Compile the MPI Program:** Use the MPI compiler wrapper to compile the program:

```
$ mpicc -o helloworld_mpi helloworld_mpi.c
```

3. **Run the MPI Program:** Execute the program with multiple processes:

```
$ mpirun -np 4 ./helloworld_mpi
```

This should produce output similar to:

```
From process 0 out of 4, Hello World!!
From process 1 out of 4, Hello World!!
From process 2 out of 4, Hello World!!
From process 3 out of 4, Hello World!!
```

Note that the order of the output lines may vary due to the parallel nature of the execution.

3 Conclusion

This report has detailed the process of setting up a virtual environment for parallel programming using C/C++ and OpenMPI. The installation process involves creating a Linux virtual machine, installing and verifying a C/C++ compiler, and installing and testing the OpenMPI library.

The successful execution of the sample MPI program confirms that the environment is correctly configured for parallel programming development. This setup provides a foundation for developing more complex parallel applications that can leverage multiple processors, potentially across multiple machines.

The virtual machine approach offers a safe and isolated environment for experimentation and development, making it an ideal solution for learning parallel programming concepts without risking system stability. As parallel computing continues to grow in importance, having

a properly configured development environment becomes increasingly valuable for students, researchers, and professionals in the computing field.

4 References

1. Isnin, I. F. (n.d.). Installing C/C++ Compiler and OpenMPI library in a linux virtual machine.
2. Ubuntu Community Help Wiki. (n.d.). InstallingCompilers. Retrieved from https://help.ubuntu.com/community/InstallingCompilers
3. Mantas, J. (n.d.). Installing Open MPI. Retrieved from http://lsi.ugr.es/jmantas/pdp/ayuda/datos/instalaciones/Install_OpenMPI_en.pdf
4. Message Passing Interface Forum. (2021). MPI: A Message-Passing Interface Standard. Retrieved from https://www.mpi-forum.org/docs/
5. Open MPI: Open Source High Performance Computing. (n.d.). Retrieved from https://www.open-mpi.org/
6. VirtualBox. (n.d.). Oracle VM VirtualBox User Manual. Retrieved from https://www.virtualbox.org/manual/