COLLEGE OF ARTS AND SCIENCES
UNIVERSITY OF THE PHILIPPINES LOS BAÑOS
1/F Wing C Physical Sciences Bldg., Harold Cuzner Royal Palm Ave.
College 4031, Laguna, Philippines
Phone +63 49 536 2313
ics.uplb@up.edu.ph www.ics.uplb.edu.ph
INSTITUTE OF COMPUTER SCIENCE

CMSC 180: Introduction to Parallel Computing

Distributed Computation of Problem via socket programming

Ariel Doria

Introduction

In larger networks of computers, computations are distributed among different computers, especially when dealing with large datasets. This allows the tasks to be distributed among available processors and increase the resources available for computations.

Contents

Learning Objectives

At the end of this discussion, the student must be able to:

- Understand what socket and socket programming is;
- Differentiate TCP and UDP;
- Enumerate the types of sockets.

Discussion

Socket programming

Socket is a communication endpoint of a two-way communication link between two programs where you can assign a name and address. Socket programming, on the other hand, is the process of how to use the socket APIs to establish communication links among the remote and local processes. Sockets are useful in stand-alone and network applications. This allows you to exchange information on the same machine or across the network, distribute work in an inefficient manner and allows access to centralized data. (*Socket Programming*, 2012)

Sockets were designed to implement the client-server model for inter-process communication where:

CMSC 180: Introduction to Parallel Computing

- The interface to network protocols needs to accommodate multiple communication protocols, such as TCP/IP, XNS, and UNIX domains.
- The interface to network protocols needs to accommodate server code that waits for connections and client code that initiates connections.
- They also need to operate differently, depending on whether the communication is connection-oriented or connectionless.
- Application programs may wish to specify the destination address of the datagrams it delivers instead of binding the address with the `open()` call. *("Programming With Sockets," 1995, #)*

Protocols

A pair of processes uses the services provided by the transport layer for communication. There are two common transport-layer protocols in the TCP/IP suite: Transmission Control Protocol and User Datagram Protocol. The choice of the transport protocol affects the capability of the application processes.

TRANSMISSION CONTROL PROTOCOL (TCP)	**USER DATAGRAM PROTOCOL (UDP)**
Requires a three-way handshake to establish a connection	Does not require to establish a connection
Guarantees the recipient will receive the packets in order; resends packets that are not received	No guarantee you are getting all the packets.
suited for applications that require high reliability, and transmission time is relatively less critical	suitable for applications that need fast, efficient transmission
commonly used in HTTP, FTP (big data files transfer), and SSH	commonly used in network diagnostics and online games

Socket Types

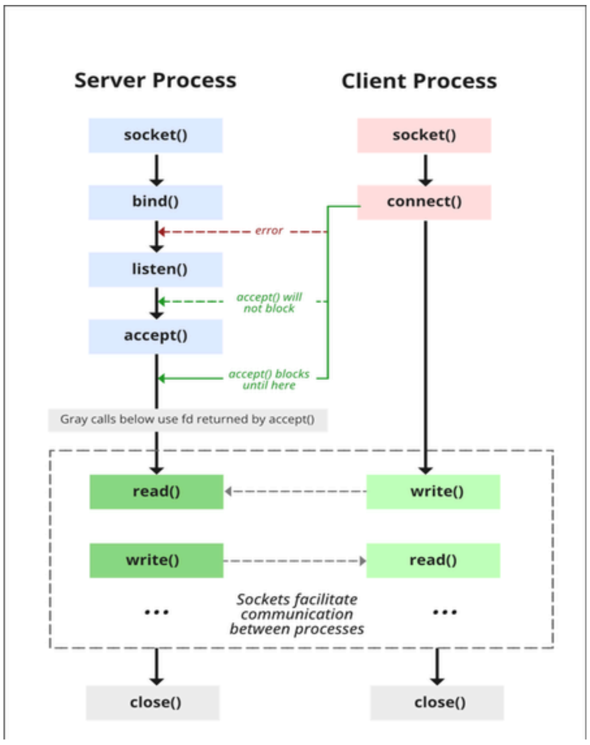
Sockets types define the communication properties visible to a user. The types of sockets are stream sockets, datagram sockets and raw sockets.

CMSC 180: Introduction to Parallel Computing

1. Stream sockets allow processes to communicate using TCP. A stream socket provides bidirectional, reliable, sequenced, and unduplicated flow of data with no record boundaries. Once the connection has been established, data can be read from and written to these sockets as a byte stream.

2. Datagram sockets allow processes to use UDP to communicate. A datagram socket supports bidirectional flow of messages. A process on a datagram socket may receive messages in a different order from the sending sequence and may receive duplicate messages. Record boundaries in the data are preserved.

3. Raw sockets provide access to Internet Control Message Protocol (ICMP). These sockets are normally datagram oriented, although their exact characteristics are dependent on the interface provided by the protocol. Raw sockets are not for most applications. They are provided to support developing new communication protocols or for access to more esoteric facilities of an existing protocol. Only superuser processes may use raw sockets. ("Programming With Sockets," 1995, #)



State diagram for server and client model of socket (*Socket Programming in C/C++*, 2022)

CMSC 180: Introduction to Parallel Computing

References

Programming With Sockets. (1995). In *Transport Interfaces Programming Guide* (pp. 11-13). https://people.cs.umass.edu/~mcorner/courses/377/socketProgramming.pdf

Socket programming. (2012). IBM. Retrieved January 20, 2023, from https://www.ibm.com/docs/en/i/7.1?topic=communications-socket-programming

Socket Programming in C/C++. (2022, September 19). GeeksforGeeks. Retrieved January 20, 2023, from https://www.geeksforgeeks.org/socket-programming-cc/

Socket programming (2018). CMSC 137 Data Communication and Networking

Online References in Socket Programming

- C
    - https://www.geeksforgeeks.org/tcp-server-client-implementation-in-c/
    - https://blogs.emorphis.com/say-hello-world-with-socket-programming-in-c/
- C++
    - https://www.geeksforgeeks.org/socket-programming-cc/
- Java
    - https://www.geeksforgeeks.org/socket-programming-in-java/
- Python
    - https://www.geeksforgeeks.org/socket-programming-python/