EC-330 COMPUTER NETWORKS

SEMESTER PROJECT
VIRTUAL CLOUD ENVIRONMENT

PROJECT REPORT

Submitted by:

USMAN KHALID, AITAZAZ KHAN

YAQOOB JAMAL

CONTENTS

Introduction to Problem

To develop a virtual cloud by joining two computers.

1. A client devices should be able to connect with the cloud server to perform different tasks.
2. An android device, which should be able to periodically log its temperature and location readings to the cloud.
3. A PC should be able to store and retrieve its files to/from the cloud.
4. Logging on from computer should give temperature and location information of the android device.
5. Since, the cloud comprises of multiple computers, files uploaded on cloud should be distributed across all computers in your cloud.

What is Cloud Computing?

Cloud computing, often referred to as simply "the cloud," is the delivery of on-demand computing resources — everything from applications to data centers — over the internet on a pay-for-use basis.

Elastic resources — Scale up or down quickly and easily to meet demand

Metered service so you only pay for what you use

Self service — All the IT resources you need with self-service access

Client-Server Data Transfer Application

The solution for proposed for the above problem comprises of a multi-client, multi-threaded server, a client application and an android application. It includes Server-Client Model. The server is implemented on C++. The client comprises of two parts, PC Client on C++ and Andriod Client on JAVA.
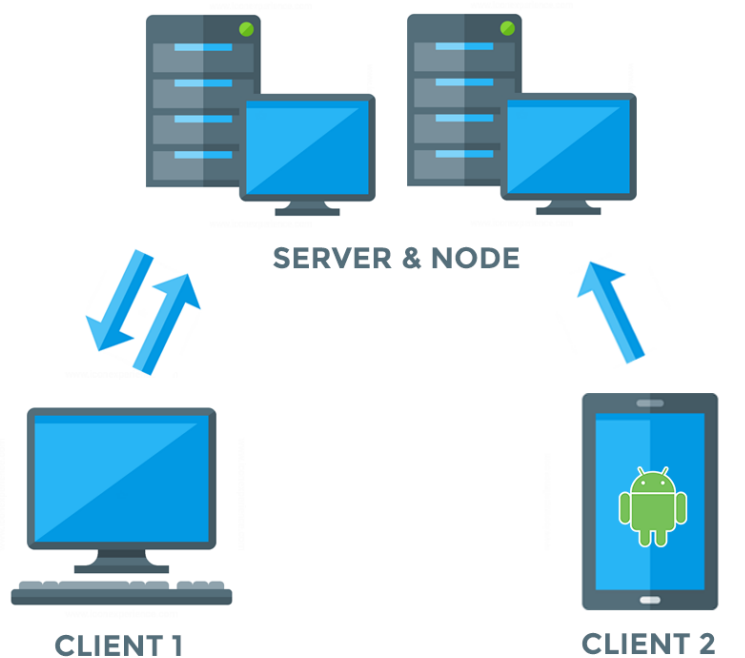
The application uses built in libraries of C++ to create TCP sockets at both Server end and Client end. The Server is running with the TCP Socket on Port Number 8888. A port number is the logical address of each application or process that uses a network or the Internet to communicate. A port number uniquely identifies a network-based application on a computer.

The Clients have to know Server's IP in order to connect with the server socket. Whenever a new client connects or exists, the server shares the information with the respective clients.

Master & Slave Server

The server listens for new connections at port 8888. There has been set a backlog of 5 clients that can connect to the Server. In order to connect multiple clients, multithreading is used using the pthread libraries.
The Server comprises of one main Server that is the 'Master Server' and a node that is the 'Slave Server'. The purpose of the Slave Server is that whenever a file is uploaded on the cloud, it is shared in between the Master Server and the Slave Server. One of the major application of cloud is elasticity of the resources. In order to ensure this, two tier Servers were implemented. The application works such, if the Master Server is not running and is crashed or some reasons, the Node Server would remain active and would help the transfer of data from the Server and the Clients. Secondly, in a case if the data in the Master Server is damaged, we would always have a backup of the data in the Slave Server.
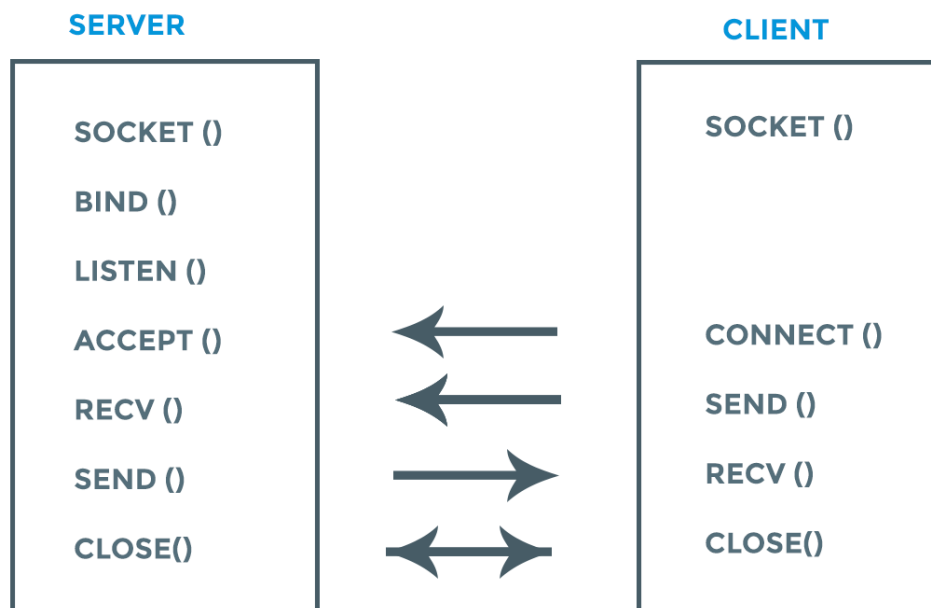
PC Client

Although the client/server model can be used by programs within a single computer, it is a more important concept of networking. In our case, the PC Client establishes a connection to a server over a TCP socket a sends data to the Server. The data can be a text file or an image. The Server also supports files of type .pdf, mp3, mp4, .exe, .cpp. The files uploaded on the main Server is shared with the Node Server. Once the server has fulfilled the client's request, the client may terminate the connection.

Android Client

One of the major feature of the cloud application is that it's access is not just limited to the PC clients. Using our application,And android device can also transfer data in the form of String. A TCP connection is first established between the Android client and the Server using Server's ip and port no. The android app sent the users information such as it's geographical location and finds the weather, temperature, humidity and pressure of that specific location using openweather API. The data is sent to the server where a log is maintained. Any client can view this log by downloading it into their PC.

Working model of Client/Server App

SERVER		CLIENT
SOCKET ()		SOCKET ()
BIND ()		
LISTEN ()		
ACCEPT ()	←	CONNECT ()
RECV ()	←	SEND ()
SEND ()	→	RECV ()
CLOSE()	↔	CLOSE()

Features of the App

1. Login Authentication

The app comprises of a login authentication. Any client connected to the server authenticates first. After three unsuccessful attempts, the server closes the socket with the client. The client has to reestablish the connection with the server in order to login to the cloud.

2. Download Files

Once the client make successful attempt to the login, the client has the access to the cloud. The client can download files from the cloud. Files may include .jpeg, .png images, .txt, .ppt, .docx text files, .exe setups, .mp3 audio or .mp4, .flv video files. There is no limit of size for file download. This is one big success of our cloud app. The transfer rate depends on the speed of the internet.

3. Upload Files

The client also has access to upload his/her own files on the cloud. The files could be accessed by all the clients connected to the server, making it a good platform for the users to socialize or exchange their data in groups. The file formats supported are the one mentioned above.

4. Backup Node

A backup server is a type of server that enables the backup of data, files, applications and/or databases on a specialized in-house or remote server. It combines hardware and software technologies that provide backup storage and retrieval services to connected computers, servers or related devices. A backup server is generally implemented in an enterprise IT environment where computing systems across an organization are connected by a network to one or more backup servers.