Design Document
Q2 - Web Server implementation

BY

SUGAM GARG 2014A7PS092P
VISHAL ARYA 2014A7PS073P
PRANAV SOOD 2014A7PS155P

PROBLEM STATEMENT:

To design a web server using pre-forking (process pool) model. Write a program named webserver.c and fastCGI.c for the following requirements. Server can accept two kinds of requests: request for .cgi files (dynamic content) and request for any "other" file type (static content).

a) Initial process pool size(I) and port number is taken as the command-line argument. Server adjusts the pool size by taking note of the free processes in the pool. The adjustment is done using the following formula. If number of free processes in the pool exceed 25% of I, kill the excess processes which have handled highest number of connections. If number of free processes go below 25% of I, add new processes to the pool to bring to 25% of I. The number of free processes is computed using a structure in shared memory. Structure is updated by processes in the pool and the parent process.

b) When web server receives a connection for either .cgi or other file types, such as .html, .pdf, .doc, or image files, this connection is taken up by one of the process in the pool. This process checks the file type. If the file type is .cgi, it will process according to steps given later. For other type of file, it will check whether that file exists/accessible and send that file to the user. Process closes the connection when client closes it or observes that no request received for last 5mins on that connection.

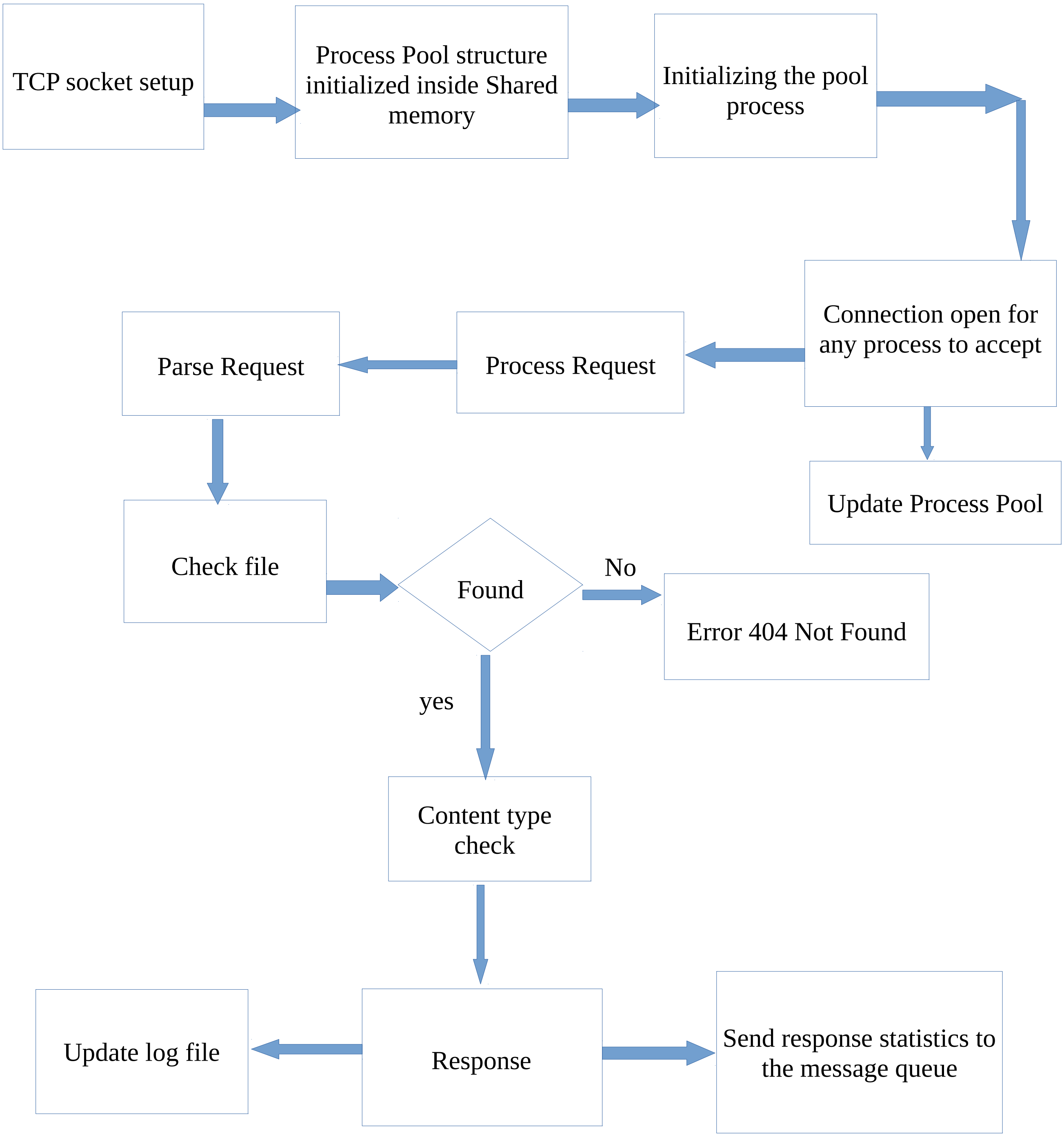
c) Web server also keeps a log file (log.csv) with columns as time stamp, user ip address, user port number, user agent, request size, query string, process time, HTTP error codes for each request received. This is for all files .cgi or other. Logging is done by the parent process which accesses these records using a message queue and writes them to log file.

d) When a request for .cgi is received, process sends this request to a FAST CGI process using full duplex pipe or TCP socket. This option is taken on CLA (-p or -t). A fast CGI process is created by each server process when .cgi request is received for the first time. A fast CGI process is a persistent process i.e. once created it stays as long as server process exists. A fast CGI process waits for the request, processes it and sends back the reply on the same connection. A server process after sending .cgi request to fast CGI process, waits for the reply. It receives the reply and sends back to the client. How to pass request to a fastCGI process is explained here. Suppose a request such as localhost/getStudentName.cgi?idno=2013A1PS999P is received. This request is sent in the following format to fastCGI process. It will execute a file named getStudentName.cgi which is a compiled C file, get the output and pass onto the server process. The protocol used in fastCGI has multiple features. Here we will go for a simple format. Server process will just forward the HTTP request received from the client to fastcgi process. fastCGI process will reply using the HTTP response format.

Program should print details of process pool creation, stats when read by parent, log entries to output.

System Description

Block Diagram:


TCP socket setup
Process Pool structure initialized inside Shared memory
Initializing the pool process
Connection open for any process to accept
Update Process Pool
Process Request
Parse Request
Check file
Found
No
Error 404 Not Found
yes
Content type check
Response
Update log file
Send response statistics to the message queue