Lab 3 Design Document

Shridhik John

Cruz ID: shjohn

CMPS 130, Fall 2019

1) Goal

The goal for Assignment 3 is to modify the HTTP server that you already implemented to have one additional feature: caching. Caching means that I am going to maintain a buffer in my server that contains a subset of the pages. When a request is received, if the requested page is in the cache, then it is read from the cache (if it is a GET request) or updated in the cache (if it is a PUT request). Otherwise, the page is first read from disk into the cache. In the log record of each request, I will indicate whether the page was in the cache at the time the request was received.

2) Assumptions

I assume I can use my code from Assignment 1, and 2. I also assume that I don't have to have multithreading functionality working. I assume I can reuse my logging features for the new assignment as well as resources provided in lab and lecture. I assume that we will be given multiple files to test whether our file can use the cache properly or not. Because of this I will primarily be using a shell.sh file to test my server.

3) Design

My general approach for this lab was to start with the code from Assignment 2. The first part I'll work on implementing is removing the multithread feature. Then I'll check the argument to check if the "-c" flag is enabled. If it is I'll go to the logging function and make sure the print statements are right. Then I'll work on implementing FIFO.

To implement FIFO, I'll first fill in the first four blocks in the cache. And then check. Once all the four blocks are full, I'll check if the next file matches any of the current files. If the cache is full and the filename doesn't match any of the current files in the cache, I'll increment a counter, and perform a modulo operation on the counter with 4. This will give me a result between 1-4, depending on the what the result is, I'll replace that number in my cache.

4) Pseudocode

```
void header(int handler, int status)
{// Check to see if file exists/is readable
        if exists send OK message
        else send appropriate Error message
}

void *connection (void *p) {
//cache memory allocation
int ref f1,f2,f3,f4;
//if response is a Get
        // Get file name
         fname = strtok(NULL, " ");
         if (fname[0] == '/') fname++;

                //Check to see if file name is exactly 27 characters, consists of a hyphen,
                underscore, or alphanumeric value values
                // Check to see if file exists and is readable
                // If fails send error message from header function
                // else continue

        send(new_socket, buf, strlen(buf), 0); // Send to client

                   //printf("%s",logbuf);
                   if (cacheenb == 1)
                   {




        // . If the cache is full and the filename doesn't match any of the current files in the cache,
I'll increment a counter, and perform a modulo operation on the counter with 5.
```

```
if ((ref_f1 == 1) && (ref_f2 == 1) && (ref_f3 == 1) && (ref_f4 == 1) &&
(strcmp(fnary0,fname)!=0) && (strcmp(fnary1,fname)!=0) &&(strcmp(fnary2,fname)!=0) &&
(strcmp(fnary3,fname)!=0) )
                        {
                          replace = counter % 5;

                          if (replace == 0)
                          {
                             ref_f1 = 0;
                             strcpy(buf1, "");
                          }
                         if (replace == 1)
                         {
                             ref_f2 = 0;
                             strcpy(buf2, "");
                         }
                         if (replace == 2)
                         {
                             ref_f3 = 0;
                             strcpy(buf3, "");
                         }
                         if (replace == 3)
                         {
                             ref_f4 = 0;
                             strcpy(buf4, "");
                         }}
                     }


                      }
                     }

         // if the there is a file in the first cache slot and cache is enabled

                     if ((ref_f1 == 1) && (cacheenb == 1))
                     {
                        if (strcmp(fnary0,fname)==0)
                        {
                           ref_f1 = 1;
         send(*new_socket_thread, buf1, strlen(buf1), 0);
                           if (logenb==1)
                           {
```

```
                    sprintf(logbuf + strlen(logbuf), "GET %s length 0[was in
chache]\n",fname);
                    sprintf(logbuf + strlen(logbuf), "========\n");
                  }
                  return(0);
                }
              }
      // if the there is no file in the first cache put the file in the first cache

              if (ref_f1 == 0)
              {
                counter++;
                fnary0 = strdup(fname);
                filed[0] = open(fname, O_RDWR);
                ref_f1 = 1;
                while(read( filed[0], buf, 1) == 1)
                {
      }


                if (logenb==1)
                {
                  if (cacheenb == 1)
                      sprintf(logbuf + strlen(logbuf), "GET %s length 0[was not in
chache]\n",fname);
                  else
                      sprintf(logbuf + strlen(logbuf), "GET %s length 0\n",fname);

                  sprintf(logbuf + strlen(logbuf), "========\n");
                }

                close(filed[0]);
               return(0);
              }


              if ((ref_f1 == 1) && (ref_f2 == 1) && (cacheenb == 1))
              {
                if (strcmp(fnary1,fname)==0)
                {
                  ref_f2 = 1;
                  send(*new_socket_thread, buf2, strlen(buf2), 0);
                  if (logenb==1)
                  {
                    sprintf(logbuf + strlen(logbuf), "GET %s length 0[was in
chache]\n",fname);
```

```
                        sprintf(logbuf + strlen(logbuf), "========\n");
                    }
                    return(0);
                }
            }
    // if the there is a file in the first cache slot but not the second

            if ((ref_f1 == 1) && (ref_f2 == 0))
            {
                counter++;
                fnary1 = strdup(fname);
                filed[1] = open(fname, O_RDWR);
                ref_f2 = 1;
                while(read( filed[1], buf, 1) == 1)
                {
    }
                if (logenb==1)
                {
                    if (cacheenb == 1)
                        sprintf(logbuf + strlen(logbuf), "GET %s length 0[was not in
chache]\n",fname);
                    else
                        sprintf(logbuf + strlen(logbuf), "GET %s length 0\n",fname);

                    sprintf(logbuf + strlen(logbuf), "========\n");
                }
                close(filed[1]);
               return(0);
            }


     // if the there is a file in the first second and third cache slot
          if ((ref_f1 == 1) && (ref_f2 == 1) && (ref_f3 == 1) && (cacheenb == 1))
            {
                if (strcmp(fnary2,fname)==0)
                {
                    ref_f3 = 1;
    send(*new_socket_thread, buf3, strlen(buf3), 0);
                    if (logenb==1)
                    {
                        sprintf(logbuf + strlen(logbuf), "GET %s length 0[was in
chache]\n",fname);
                        sprintf(logbuf + strlen(logbuf), "========\n");
                    }
                    return(0);
                }
```

```
                    }

        // if the there is a file in the first cache slot and the second, nut not the third

                    if ((ref_f1 == 1) && (ref_f2 == 1) && (ref_f3 == 0))
                    {
                      counter++;
                      fnary2 = strdup(fname);
                      filed[2] = open(fname, O_RDWR);
                      ref_f3 = 1;

                      while(read( filed[2], buf, 1) == 1)
                      {
        }
                      if (logenb==1)
                      {
                        if (cacheenb == 1)
                              sprintf(logbuf + strlen(logbuf), "GET %s length 0[was not in
chache]\n",fname);
                        else
                              sprintf(logbuf + strlen(logbuf), "GET %s length 0\n",fname);

                        sprintf(logbuf + strlen(logbuf), "========\n");
                      }
                      close(filed[2]);
                      return(0);
                    }



                // if the there is a file in the first cache slot, second, third, and fourth

                    if ((ref_f1 == 1) && (ref_f2 == 1) && (ref_f3 == 1) && (ref_f4 == 1) &&
(cacheenb == 1))
                    {
                      if (strcmp(fnary3,fname)==0)
                      {
                        ref_f4 = 1;
        send(*new_socket_thread, buf4, strlen(buf4), 0);
                        if (logenb==1)
                        {
                          sprintf(logbuf + strlen(logbuf), "GET %s length 0[was in
chache]\n",fname);
                          sprintf(logbuf + strlen(logbuf), "========\n");
                        }
                        return(0);
```

```
                            }
                        }

                    // if the there is a file in the first cache slot, second, third, but not the fourth fourth

                        if ((ref_f1 == 1) && (ref_f2 == 1) && (ref_f3 == 1) && (ref_f4 == 0))
                        {
                            counter++;
                            fnary3 = strdup(fname);
                            filed[3] = open(fname, O_RDWR);
                            ref_f4 = 1;
                            while(read( filed[3], buf, 1) == 1)
                            {
        }
                            close(filed[3]);
                            if (logenb==1)
                            {
                                if (cacheenb == 1)
                                        sprintf(logbuf + strlen(logbuf), "GET %s length 0[was not in
chache]\n",fname);
                                else
                                        sprintf(logbuf + strlen(logbuf), "GET %s length 0\n",fname);

                                sprintf(logbuf + strlen(logbuf), "========\n");
                            }
        return(0);
                        }
                    }
            }

//if response is a Put
        // Get file name
         fname = strtok(NULL, " ");
         if (fname[0] == '/') fname++;
        // . If the cache is full and the filename doesn't match any of the current files in the cache,
I'll increment a counter, and perform a modulo operation on the counter with 5.




if ((ref_f1 == 1) && (ref_f2 == 1) && (ref_f3 == 1) && (ref_f4 == 1) &&
(strcmp(fnary0,fname)!=0) && (strcmp(fnary1,fname)!=0) &&(strcmp(fnary2,fname)!=0) &&
(strcmp(fnary3,fname)!=0) )
                            {
```

```
                        replace = counter % 5;

                        if (replace == 0)
                        {
                            ref_f1 = 0;
                            strcpy(buf1, "");
                        }
                       if (replace == 1)
                       {
                            ref_f2 = 0;
                            strcpy(buf2, "");
                       }
                       if (replace == 2)
                       {
                            ref_f3 = 0;
                            strcpy(buf3, "");
                       }
                       if (replace == 3)
                       {
                            ref_f4 = 0;
                            strcpy(buf4, "");
                       }}
                  }


                   }
                  }

        // if the there is a file in the first cache slot and cache is enabled

                  if ((ref_f1 == 1) && (cacheenb == 1))
                  {
                     if (strcmp(fnary0,fname)==0)
                     {
                        ref_f1 = 1;
        send(*new_socket_thread, buf1, strlen(buf1), 0);
                        if (logenb==1)
                        {
                            sprintf(logbuf + strlen(logbuf), "GET %s length 0[was in
chache]\n",fname);
                            sprintf(logbuf + strlen(logbuf), "========\n");
                        }
                        return(0);
                     }
                  }
        // if the there is no file in the first cache put the file in the first cache
```

```
                    if (ref_f1 == 0)
                    {
                       counter++;
                       fnary0 = strdup(fname);
                       filed[0] = open(fname, O_RDWR);
                       ref_f1 = 1;
                       while(read( filed[0], buf, 1) == 1)
                       {
            }


                       if (logenb==1)
                       {
                          if (cacheenb == 1)
                                sprintf(logbuf + strlen(logbuf), "GET %s length 0[was not in
chache]\n",fname);
                          else
                                sprintf(logbuf + strlen(logbuf), "GET %s length 0\n",fname);

                          sprintf(logbuf + strlen(logbuf), "========\n");
                       }

                       close(filed[0]);
                     return(0);
                    }


                    if ((ref_f1 == 1) && (ref_f2 == 1) && (cacheenb == 1))
                    {
                       if (strcmp(fnary1,fname)==0)
                       {
                          ref_f2 = 1;
                          send(*new_socket_thread, buf2, strlen(buf2), 0);
                          if (logenb==1)
                          {
                             sprintf(logbuf + strlen(logbuf), "GET %s length 0[was in
chache]\n",fname);
                             sprintf(logbuf + strlen(logbuf), "========\n");
                          }
                          return(0);
                       }
                    }
            // if the there is a file in the first cache slot but not the second

                    if ((ref_f1 == 1) && (ref_f2 == 0))
                    {
                       counter++;
                       fnary1 = strdup(fname);
                       filed[1] = open(fname, O_RDWR);
                       ref_f2 = 1;
                       while(read( filed[1], buf, 1) == 1)
                       {
          }
                       if (logenb==1)
                       {
                          if (cacheenb == 1)
                                sprintf(logbuf + strlen(logbuf), "GET %s length 0[was not in
chache]\n",fname);
                          else
                                sprintf(logbuf + strlen(logbuf), "GET %s length 0\n",fname);

                          sprintf(logbuf + strlen(logbuf), "========\n");
                       }
                       close(filed[1]);
                      return(0);
                    }


           // if the there is a file in the first second and third cache slot
                  if ((ref_f1 == 1) && (ref_f2 == 1) && (ref_f3 == 1) && (cacheenb == 1))
                    {
                       if (strcmp(fnary2,fname)==0)
                       {
                          ref_f3 = 1;
          send(*new_socket_thread, buf3, strlen(buf3), 0);
                          if (logenb==1)
                          {
                             sprintf(logbuf + strlen(logbuf), "GET %s length 0[was in
chache]\n",fname);
                             sprintf(logbuf + strlen(logbuf), "========\n");
                          }
                          return(0);
                       }
                    }

          // if the there is a file in the first cache slot and the second, nut not the third

                    if ((ref_f1 == 1) && (ref_f2 == 1) && (ref_f3 == 0))
                    {
                       counter++;
                       fnary2 = strdup(fname);
                    filed[2] = open(fname, O_RDWR);
                    ref_f3 = 1;

                    while(read( filed[2], buf, 1) == 1)
                    {
          }
                    if (logenb==1)
                    {
                      if (cacheenb == 1)
                            sprintf(logbuf + strlen(logbuf), "GET %s length 0[was not in
chache]\n",fname);
                      else
                            sprintf(logbuf + strlen(logbuf), "GET %s length 0\n",fname);

                      sprintf(logbuf + strlen(logbuf), "========\n");
                    }
                    close(filed[2]);
                    return(0);
                  }



                // if the there is a file in the first cache slot, second, third, and fourth

                  if ((ref_f1 == 1) && (ref_f2 == 1) && (ref_f3 == 1) && (ref_f4 == 1) &&
(cacheenb == 1))
                  {
                    if (strcmp(fnary3,fname)==0)
                    {
                      ref_f4 = 1;
          send(*new_socket_thread, buf4, strlen(buf4), 0);
                      if (logenb==1)
                      {
                        sprintf(logbuf + strlen(logbuf), "GET %s length 0[was in
chache]\n",fname);
                        sprintf(logbuf + strlen(logbuf), "========\n");
                      }
                      return(0);
                    }
                  }

                // if the there is a file in the first cache slot, second, third, but not the fourth fourth

                  if ((ref_f1 == 1) && (ref_f2 == 1) && (ref_f3 == 1) && (ref_f4 == 0))
                  {
                    counter++;
```

```
                        fnary3 = strdup(fname);
                        filed[3] = open(fname, O_RDWR);
                        ref_f4 = 1;
                        while(read( filed[3], buf, 1) == 1)
                        {
        }
                        close(filed[3]);
                        if (logenb==1)
                        {
                          if (cacheenb == 1)
                                sprintf(logbuf + strlen(logbuf), "GET %s length 0[was not in
chache]\n",fname);
                          else
                                sprintf(logbuf + strlen(logbuf), "GET %s length 0\n",fname);

                          sprintf(logbuf + strlen(logbuf), "========\n");
                        }
        return(0);
                      }
                    }
          }

// Open a file and read/Write with this line command from Dog.c
        fd = open(fname, O_CREAT | O_WRONLY | O_TRUNC);
        recv(new_socket, buf, 50, 0);
        write( fd, buf, 50);
close(fd); // close file
```

int Main(int argc, char const *argv[])

```
{

        // go through arguments and find "-l"
        if (strcmp((char *) argv[opt], "-l")==0){
                 logenb = 1;
                // the filename for the log file is the name after -l
                filename = (char *) argv[opt + 1];
      }
        // go through the arguments and find –c to check if we're enabling the cache or not
      if (strcmp((char *) argv[opt], "-c")==0){
        cacheenb = 1;
      }
        // go through the arguments and find –N to check how many arguments we're being
given
      if (strcmp((char *) argv[opt], "-N")==0){
        nf = (char *) argv[opt + 1];
        numfile = atoi(nf);
      }
  }
  }
```

insert Geek for Geeks code to set up socket

5) Question

• Using your new httpserver with caching, perform an experiment to demonstrate how caching can improve performance (latency and/or throughput). Do a test with caching turned on and compare it with the same test but with caching turned off.

The experiment I chose to do to demonstrate caching was by using the “time” command while running my program. Caching made my performance more optimal and the results came out faster when caching was turned on.