Worksheet Class 5: HTTP Performance, Concurrent Web Servers and Cookies

Q1. How would you use threads to achieve faster performance in the following programs?

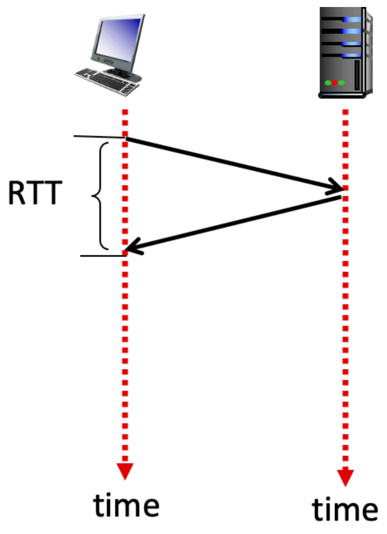
A. A program that displays the squares of numbers 1 to 10 (in any order).

B. A program that serves data to multiple clients.

C. A program that reads data from files and sends them over the network.

D. Program that reads files from the disk and prints the output.

Q2. RTT (Round Trip Time) Estimation



Round Trip Time (RTT):

- time for a small packet to travel from client to server and response to come back.
- Connection establishment (via TCP) requires one RTT.

Non-Persistent HTTP Connections can download a website with several objects in…

A. One RTT + (File transfer time per object)

B. (One RTT + File transfer time) per object

C. Two RTTs

D. Two RTTs + (File transfer time per object)

E. (Two RTTS + File transfer time) per object

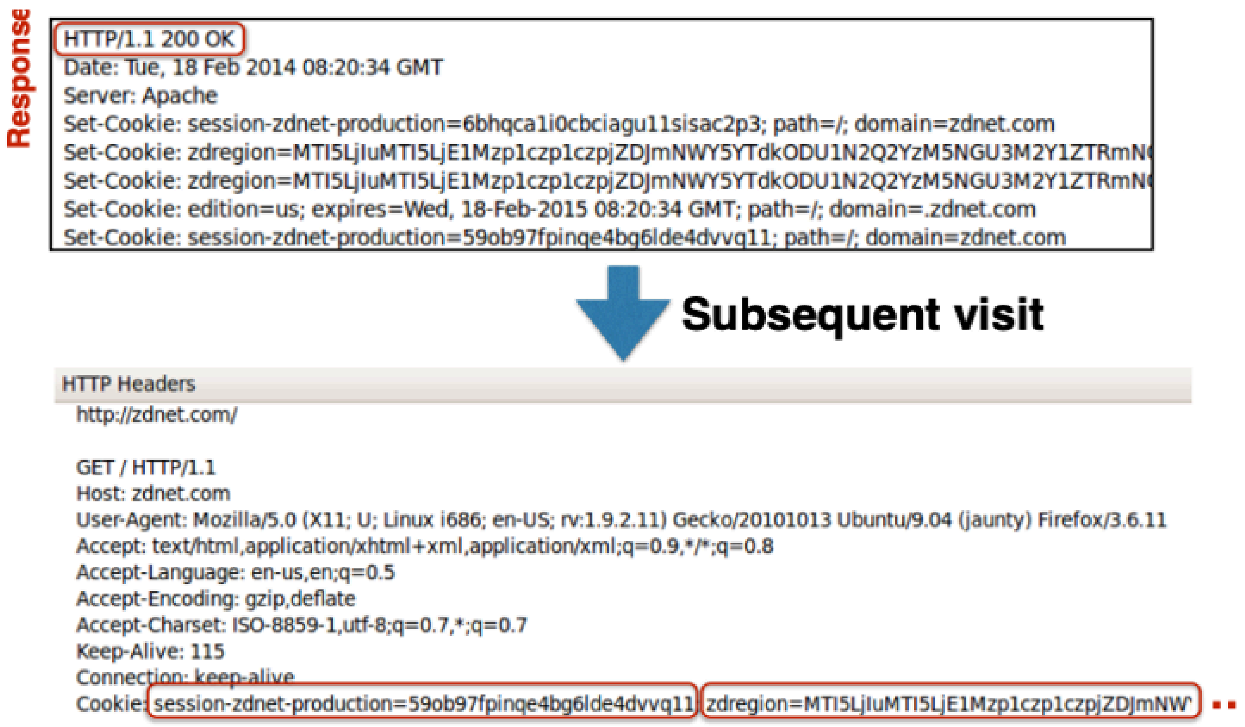
HTTP Cookies

Q1. What do we mean when we say “HTTP is stateless”? In answering this question, assume that cookies are not used. Check all answers that apply.

A. An HTTP *server* does not remember anything about what happened during earlier steps in interacting with this HTTP client.
B. An HTTP *client* does not remember anything about what happened during earlier steps in interacting with any HTTP server.
C. An HTTP client does not remember the identities of the servers with which it has interacted.
D. We say this when an HTTP server is not operational.
E. The HTTP protocol is not licensed in any country.

Q2. What is an HTTP cookie used for?

A. A cookie is used to spoof client identity to an HTTP server.
B. A cookie is a piece of code used by a server, carried on a client’s HTTP request, to access information the server had earlier stored about an earlier interaction with this *person.* [Think about the distinction between a *browser* and a *person*.]
C. A cookie is a code used by a client to authenticate a person’s identity to an HTTP server.
D. A cookie is a code used by a server, carried on a client’s HTTP request, to access information the server had earlier stored about an earlier interaction with this Web *browser*. [Think about the distinction between a *browser* and a *person*.]
E. Like dessert, cookies are used at the end of a transaction, to indicate the end of the transaction.



```
HTTP/1.1 200 OK
Date: Tue, 18 Feb 2014 08:20:34 GMT
Server: Apache
Set-Cookie: session-zdnet-production=6bhqca1i0cbciagu11sisac2p3; path=/; domain=zdnet.com
Set-Cookie: zdregion=MTI5LjIuMTI5LjE1Mzp1czp1czpjZDJmNWY5YTdkODU1N2Q2YzM5NGU3M2Y1ZTRmN
Set-Cookie: zdregion=MTI5LjIuMTI5LjE1Mzp1czp1czpjZDJmNWY5YTdkODU1N2Q2YzM5NGU3M2Y1ZTRmN
Set-Cookie: edition=us; expires=Wed, 18-Feb-2015 08:20:34 GMT; path=/; domain=.zdnet.com
Set-Cookie: session-zdnet-production=59ob97fpinqe4bg6lde4dvvq11; path=/; domain=zdnet.com
```

```
http://zdnet.com/

GET / HTTP/1.1
Host: zdnet.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.2.11) Gecko/20101013 Ubuntu/9.04 (jaunty) Firefox/3.6.11
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Cookie: session-zdnet-production=59ob97fpinqe4bg6lde4dvvq11 zdregion=MTI5LjIuMTI5LjE1Mzp1czp1czpjZDJmNW ..
```

What is the purpose of a cookie value in the HTTP GET request/response above?

A. The cookie value encodes a default set of preferences that the user has previously specified for this web site.

B. The cookie value itself doesn't mean anything. It is just a value that was returned by a web server to this client during an earlier interaction.

C. The cookie value indicates whether the user wants to use HTTP/1, HTTP/1.1, or HTTP/2 for this GET request.

D. The cookie value encodes the format of the reply preferred by the client in the response to this GET request.

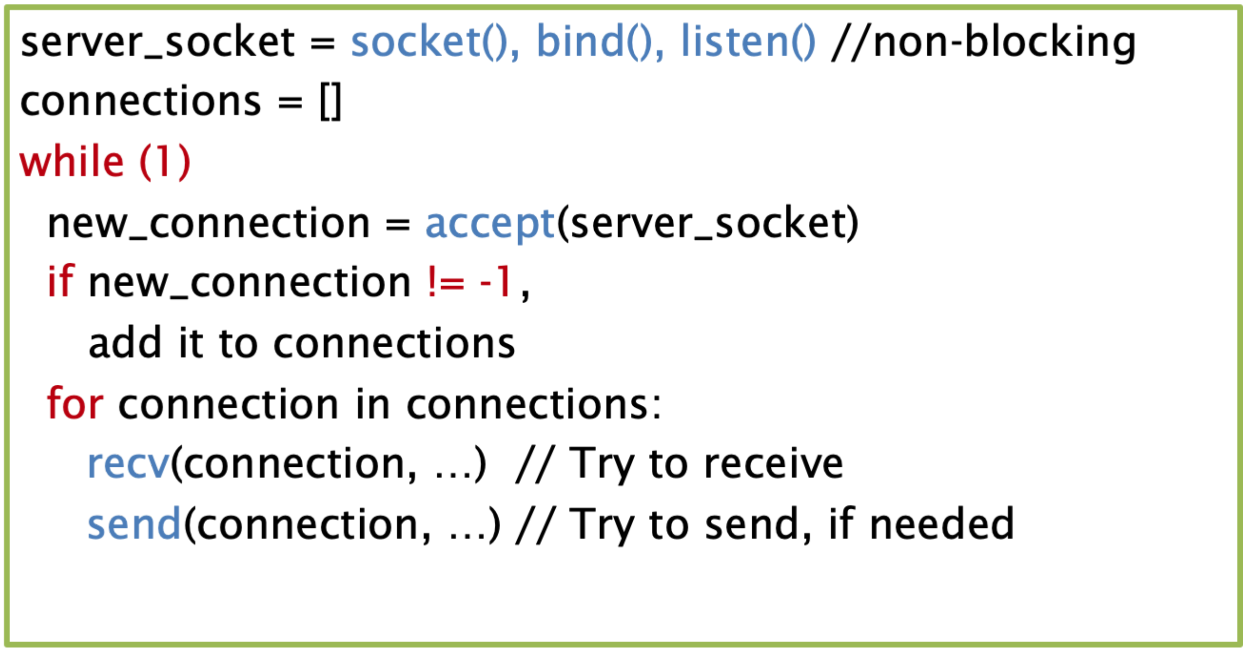
E. The cookie value is an encoding of a user email address associated with the GET request.

Concurrency in Web Servers

Q1. Which benefit of threads is the most critical in the context of running a web server?

A. Modular code/separation of concerns.
B. Multiple CPU/core parallelism.
C. I/O overlapping.
D. Some other benefit

Q2. Event-Driven Concurrency. Consider the following pseudo-code:



```
server_socket = socket(), bind(), listen() //non-blocking
connections = []
while (1)
  new_connection = accept(server_socket)
  if new_connection != -1,
     add it to connections
  for connection in connections:
     recv(connection, ...)  // Try to receive
     send(connection, ...) // Try to send, if needed
```

Which of the following options would best describe the code's performance?

A. Yes, this will work efficiently.
B. Yes but this will execute too slowly.
C. Yes but this will use too many resources.
D. No, this will still block.