Raghu Tirumala
CSS 432 Networking
Program 1: Sockets

Documentation of Algorithm:

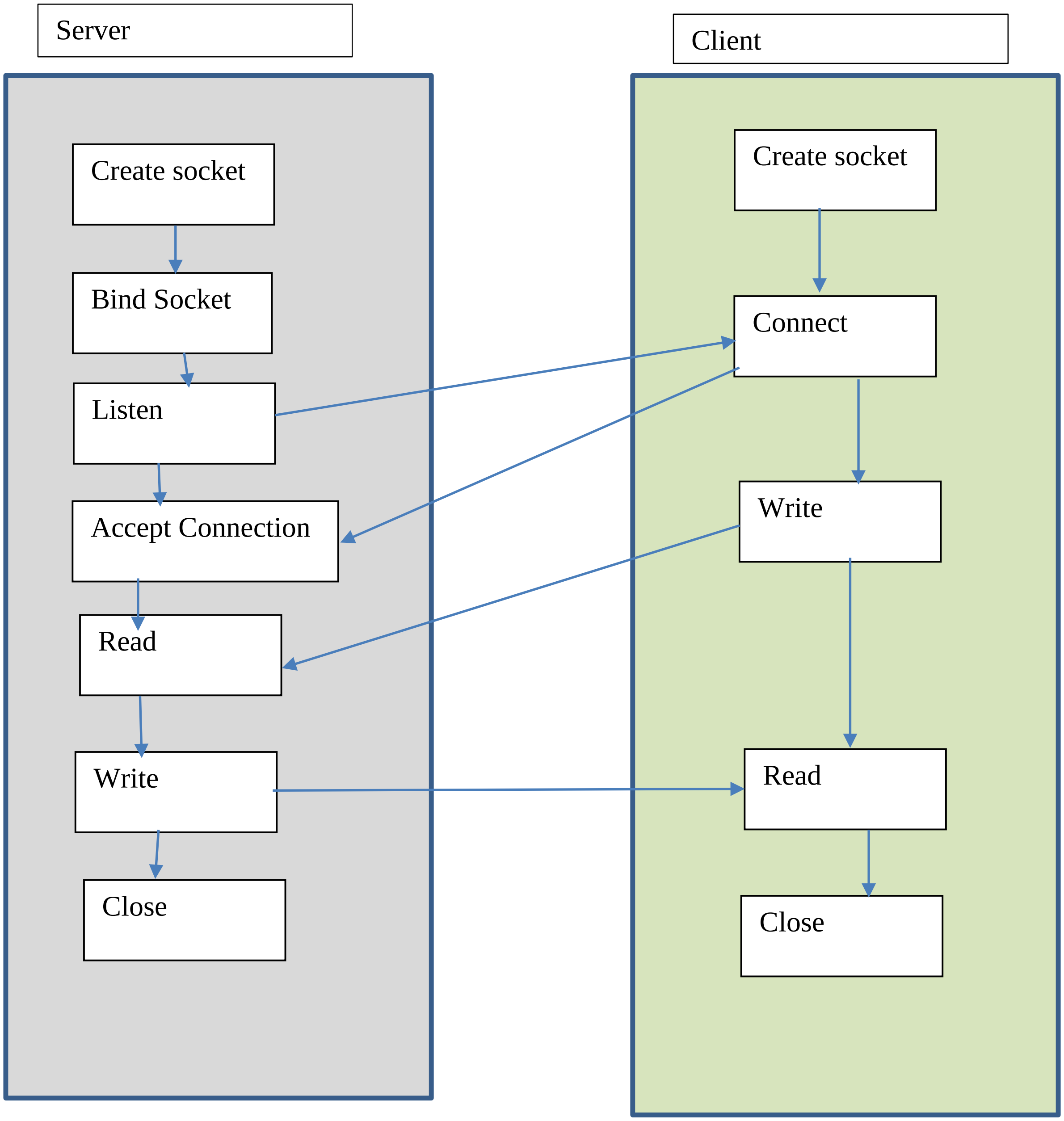
This program uses a connection-oriented socket to communicate between client and server processes.

Client.cpp

1. Receive a Server's IP port, name, number of data buffers to send, size of each databuffer, number of repetitions to write data, and the test type.
2. Open a tcp socket.
3. Connect socket to server by calling connect() function.
4. Write data to server using write or writev function calls (Repetition times where Repetition is a command line argument).
   a. If test type = 1 invoke the write() system call for each data buffer
   b. If test type = 2 allocate an array of iovec structs each having iov_base field pointing to different data buffer and iov_len storing buffer size. Then calls writev() to write to all databuffers at once.
   c. Allocates an nbufs-sized array of data buffers, and thereafter calls write( ) to send this array, (i.e., all data buffers) at once.
5. Receive from the server an integer acknowledge of how many times the server has called read.
6. Print total time of data-sending and round trip time to console.
7. Close the socket and terminate the program.

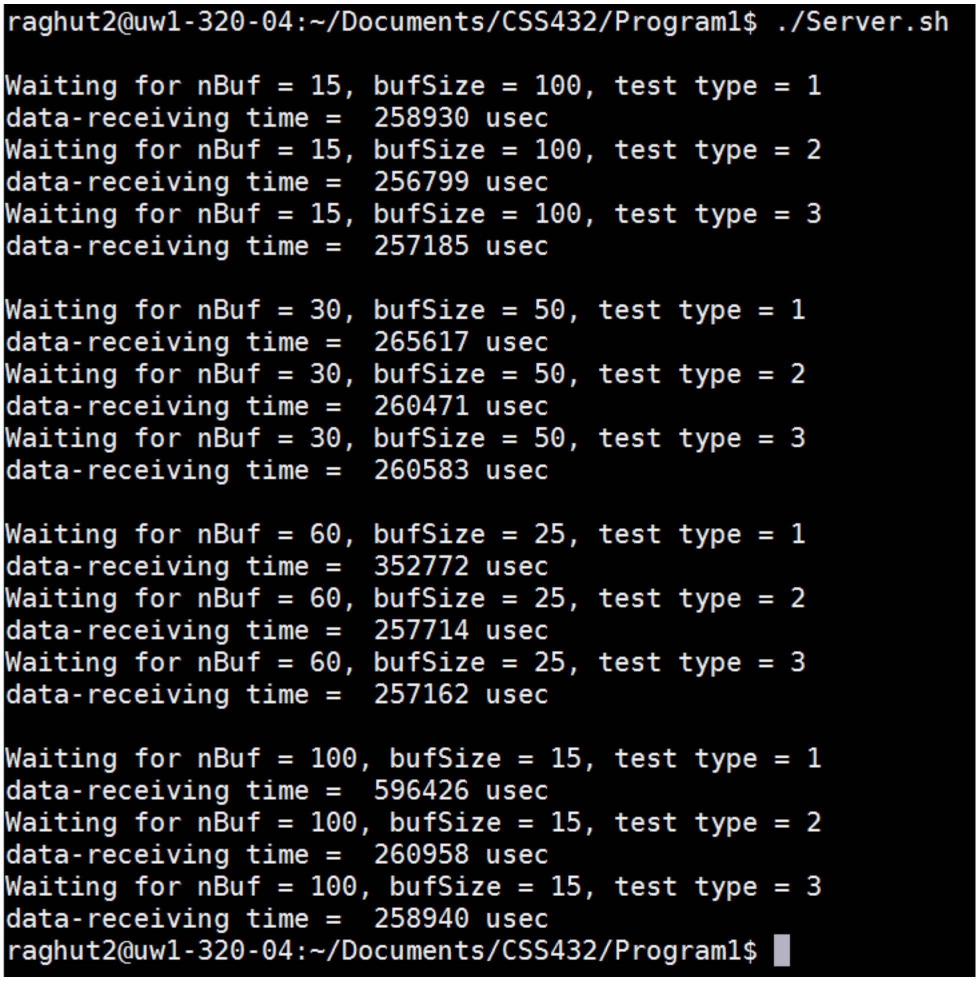
Server.cpp

1. Receive Server Port and repetitions to read data as command line arguments.
2. Create a socket.
3. Bind to its local address.
4. Listen to up to five connection requests from clients.
5. Receive a request from client by calling accept and create new socket for this specific connection.
6. Read data from the client using read() system call.
7. Write back number of reads to client as an acknowledge.
8. Close the sockets and terminate the program.


Server
Create socket
Bind Socket
Listen
Accept Connection
Read
Write
Close
Client
Create socket
Connect
Write
Read
Close

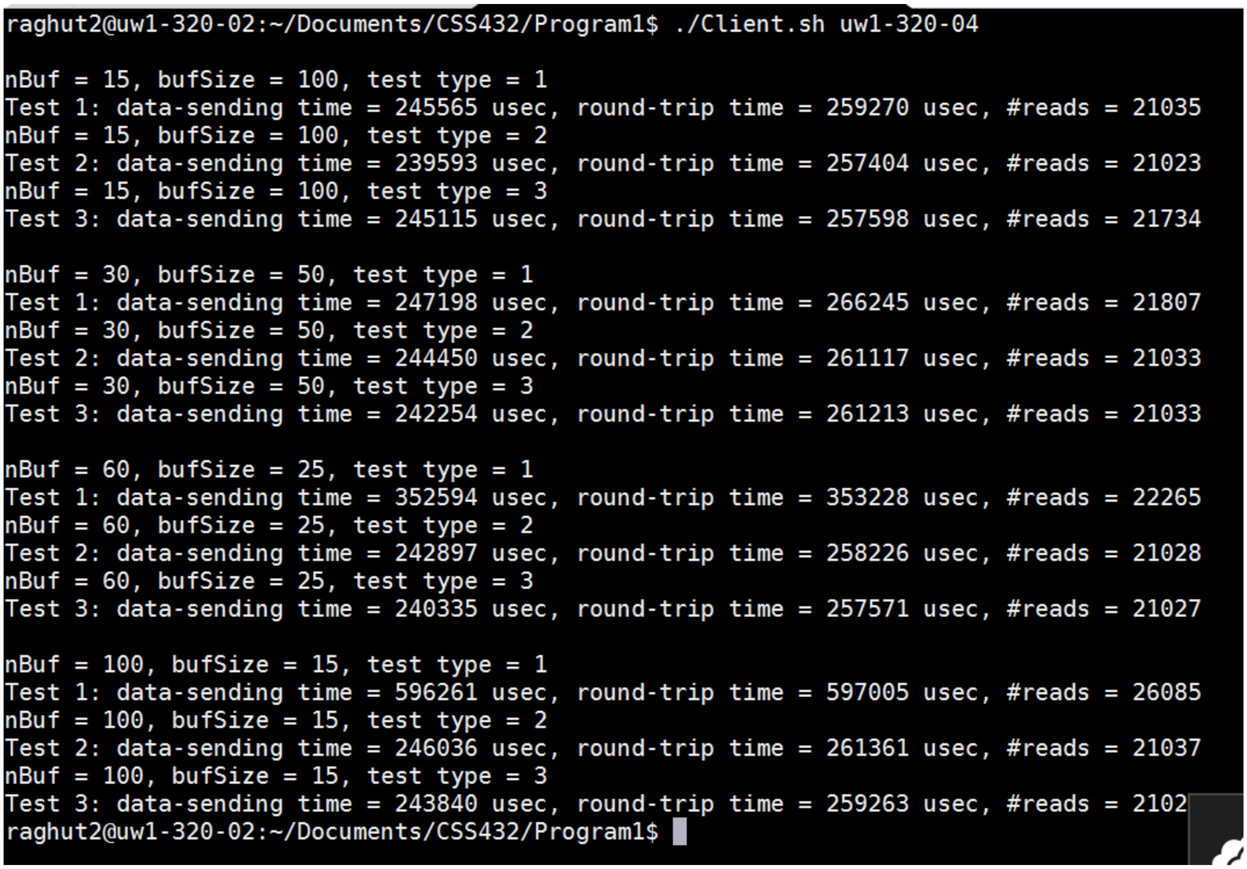
Execution Output

Server.cpp



```
raghut2@uw1-320-04:~/Documents/CSS432/Program1$ ./Server.sh

Waiting for nBuf = 15, bufSize = 100, test type = 1
data-receiving time =  258930 usec
Waiting for nBuf = 15, bufSize = 100, test type = 2
data-receiving time =  256799 usec
Waiting for nBuf = 15, bufSize = 100, test type = 3
data-receiving time =  257185 usec

Waiting for nBuf = 30, bufSize = 50, test type = 1
data-receiving time =  265617 usec
Waiting for nBuf = 30, bufSize = 50, test type = 2
data-receiving time =  260471 usec
Waiting for nBuf = 30, bufSize = 50, test type = 3
data-receiving time =  260583 usec

Waiting for nBuf = 60, bufSize = 25, test type = 1
data-receiving time =  352772 usec
Waiting for nBuf = 60, bufSize = 25, test type = 2
data-receiving time =  257714 usec
Waiting for nBuf = 60, bufSize = 25, test type = 3
data-receiving time =  257162 usec

Waiting for nBuf = 100, bufSize = 15, test type = 1
data-receiving time =  596426 usec
Waiting for nBuf = 100, bufSize = 15, test type = 2
data-receiving time =  260958 usec
Waiting for nBuf = 100, bufSize = 15, test type = 3
data-receiving time =  258940 usec
raghut2@uw1-320-04:~/Documents/CSS432/Program1$ █
```

Client.cpp



```
raghut2@uw1-320-02:~/Documents/CSS432/Program1$ ./Client.sh uw1-320-04

nBuf = 15, bufSize = 100, test type = 1
Test 1: data-sending time = 245565 usec, round-trip time = 259270 usec, #reads = 21035
nBuf = 15, bufSize = 100, test type = 2
Test 2: data-sending time = 239593 usec, round-trip time = 257404 usec, #reads = 21023
nBuf = 15, bufSize = 100, test type = 3
Test 3: data-sending time = 245115 usec, round-trip time = 257598 usec, #reads = 21734

nBuf = 30, bufSize = 50, test type = 1
Test 1: data-sending time = 247198 usec, round-trip time = 266245 usec, #reads = 21807
nBuf = 30, bufSize = 50, test type = 2
Test 2: data-sending time = 244450 usec, round-trip time = 261117 usec, #reads = 21033
nBuf = 30, bufSize = 50, test type = 3
Test 3: data-sending time = 242254 usec, round-trip time = 261213 usec, #reads = 21033

nBuf = 60, bufSize = 25, test type = 1
Test 1: data-sending time = 352594 usec, round-trip time = 353228 usec, #reads = 22265
nBuf = 60, bufSize = 25, test type = 2
Test 2: data-sending time = 242897 usec, round-trip time = 258226 usec, #reads = 21028
nBuf = 60, bufSize = 25, test type = 3
Test 3: data-sending time = 240335 usec, round-trip time = 257571 usec, #reads = 21027

nBuf = 100, bufSize = 15, test type = 1
Test 1: data-sending time = 596261 usec, round-trip time = 597005 usec, #reads = 26085
nBuf = 100, bufSize = 15, test type = 2
Test 2: data-sending time = 246036 usec, round-trip time = 261361 usec, #reads = 21037
nBuf = 100, bufSize = 15, test type = 3
Test 3: data-sending time = 243840 usec, round-trip time = 259263 usec, #reads = 2102
raghut2@uw1-320-02:~/Documents/CSS432/Program1$ 
```

Performance Evaluation

Server Data:

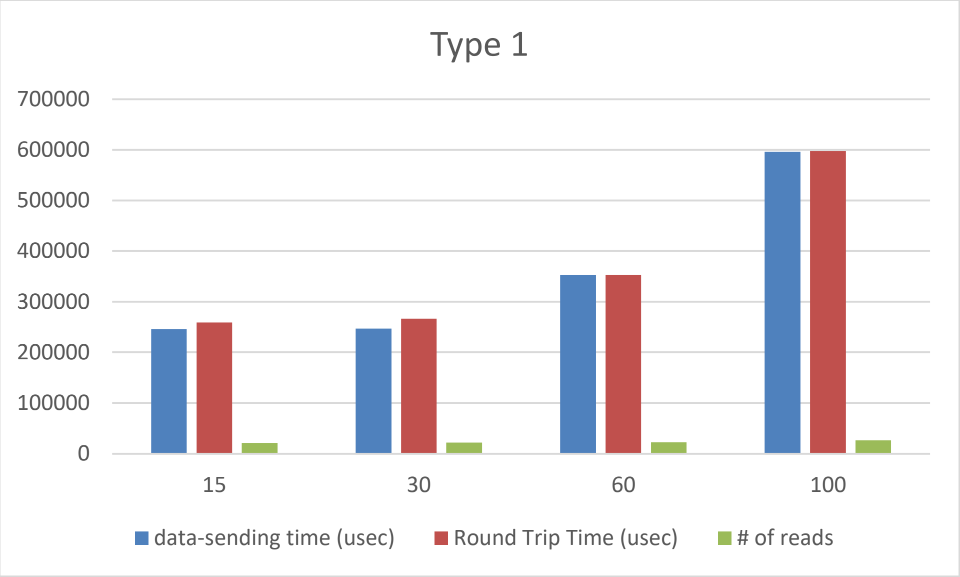
nBuf	bufSize	test type	data-receiving time (usec)	# of reads
15	100	1	258930	21035
15	100	2	256799	21023
15	100	3	257185	21734
30	50	1	265617	21807
30	50	2	260471	21033
30	50	3	260583	21033
60	25	1	352772	22265
60	25	2	257714	21028
60	25	3	257162	21027
100	15	1	596426	26085
100	15	2	260958	21037
100	15	3	258940	21028

Client Data:

nBuf	bufSize	test type	data-sending-time (usec)	round trip time (usec)
15	100	1	245565	259270
15	100	2	239593	257404
15	100	3	245115	257598
30	50	1	247198	266245
30	50	2	244450	261117
30	50	3	242254	261213
60	25	1	352594	353228
60	25	2	242897	258266
60	25	3	240335	257571
100	15	1	596261	597005
100	15	2	246036	261361
100	15	3	243840	259263

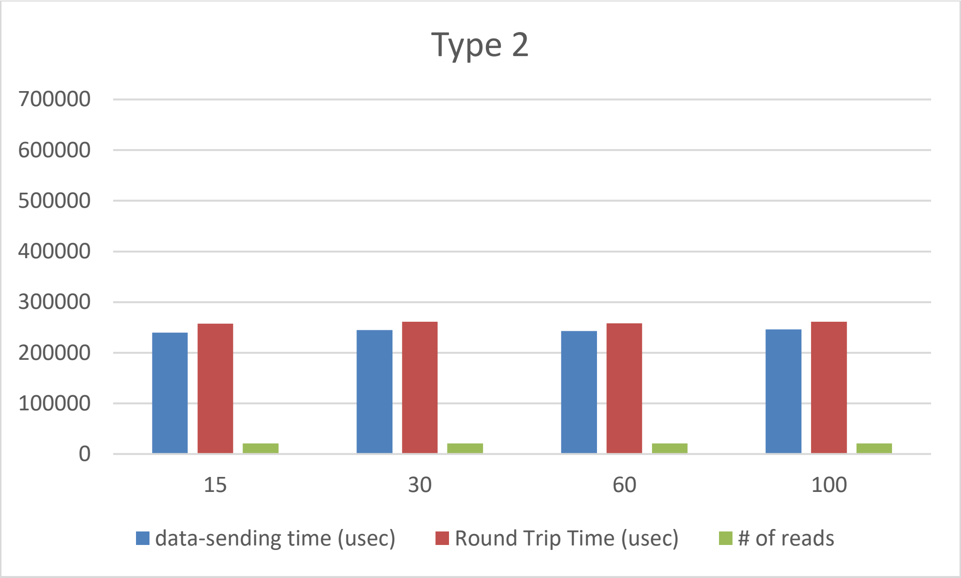
Write type 1:

nBuf	data-sending time (usec)	Round Trip Time (usec)	# of reads
15	245565	259270	21035
30	247198	266245	21807
60	352594	353228	22265
100	596261	597005	26085



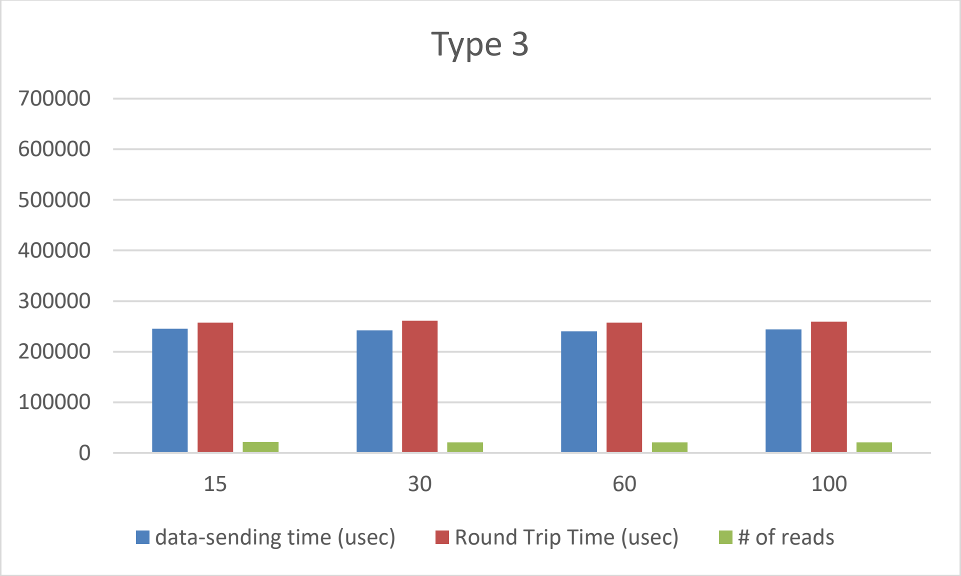
Write type 2:

nBuf	data-sending time (usec)	Round Trip Time (usec)	# of reads
15	239593	257404	21023
30	244450	261117	21033
60	242897	258266	21028
100	246036	261361	21037



Write type 3:

nBuf	data-sending time (usec)	Round Trip Time (usec)	# of reads
15	245115	257598	21734
30	242254	261213	21033
60	240335	257571	21027
100	243840	259263	21028



Discussion

This data was taken when Client.cpp was run on machine number 2 in the linux lab and Server.cpp was run on machine number 4.

Test type 1 showed the most variance in data transfer time and number of reads with differing number of buffers. There was a clear positive correlation between number of buffers (nBufs) and data transfer and number of reads. (Greater value of nBufs greater value of everything else). Write types 2 and 3 (writev and single write) seemed to have a constant number of reads and data transfer time regardless of the number of buffers.

Conclusion

Since write type 1 calls as many writes as databuffers that exist, when the number of data buffers increases, the number of write calls increases causing it to take longer to send data. Write types 2 and 3 write all the data buffers at once so do not change the time required to send data as the number of data buffers increases.