

Colt Campbell
CSCE 313_504
October 5, 2013

Report

Performance of client:

Considering the client file creates a separate process to emulate a dataserer and inter-process communication happens, the overall performance of the program is pretty decent at around 100 microseconds per request from the dataserer. For the most part, the low-level functions of `cwrite` and `cread` certainly help reduce the cost of the inter-process communication by calling functions that are as low level as possible to take input and produce output. Another plus of the client is that it will simply exit if there are any problems with either creating the new process or when loading the dataserer. The error checking may impact the program run time very slightly for the worse, but it's just a few lines, and it is necessary.

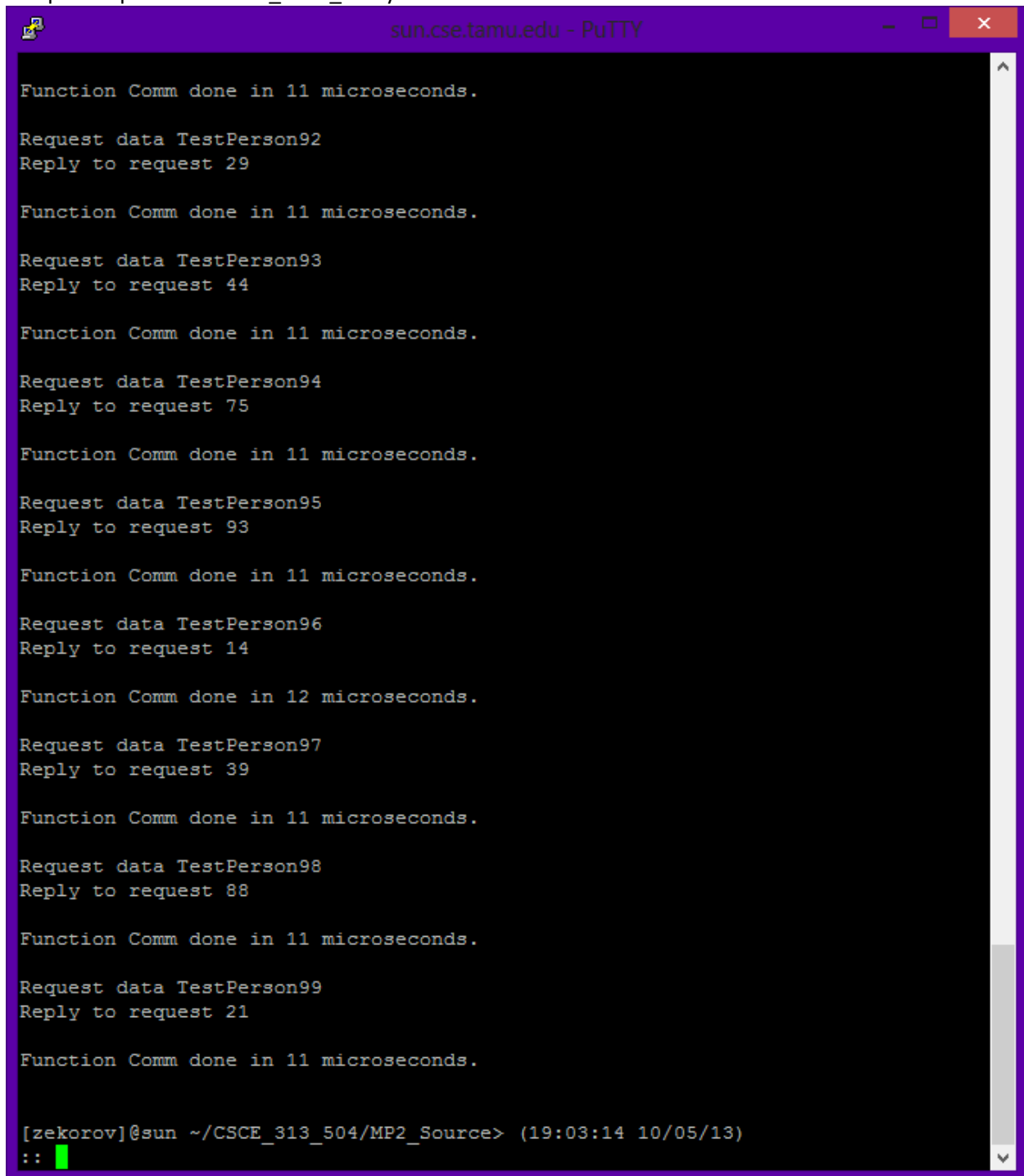
Delay time difference between local function and separate processes:

However, if we take our same client program and implement a small function that does the same thing that the client/dataserer does with requests (takes one in and then returns a reply), we can get the performance to be significantly faster than for the client/dataserer system. Essentially what is going on is that the client/dataserer system has to communicate across processes, and that makes it more resource intensive compared to the intensity of a local function in one process performing the same action the client/dataserer would across processes. The reason it is so much faster inside the local function is that for one thing, there is much less code that is required (depending on exactly how you implement the request processing since you can basically grab the `cwrite` and `cread` and process functions in the dataserer and just take the channel material out of that code, and then use it to get a local function, or a very simple function like the `process_requests_no_channels` (string request) in my `function_time_delay.C` program can be used that just returns a reply). For another thing, the function is operating within the same memory address space, and thus it is easier and faster for it to access whatever is needed to input/output the requests and replies. The client/dataserer system has to communicate across processes. One process sends a request and the other process sends the reply back to the first process. This means that the processes have to access different address space other than what they thrive in individually. Going to further address spaces requires more time, and thus makes the inter-process communication slower than the communication within a single process function as the single process operates in its own address space. The function processing time takes generally around 11 microseconds (significantly faster) whereas the client processing time takes around 100 microseconds with slightly more variation than the function processing times.

Sample Output for Client.C:

```
sun.cse.tamu.edu - PuTTY
New request is data TestPerson46
Reading next request from channel (control) ...reply to request 46:62
Process Comm done in 102 microseconds.
done (control).
New request is data TestPerson47
Reading next request from channel (control) ...reply to request 47:34
Process Comm done in 100 microseconds.
done (control).
New request is data TestPerson48
Reading next request from channel (control) ...reply to request 48:53
Process Comm done in 102 microseconds.
done (control).
New request is data TestPerson49
Reading next request from channel (control) ...reply to request 49:20
Process Comm done in 100 microseconds.
done (control).
New request is data TestPerson50
Reading next request from channel (control) ...reply to request 50:44
Process Comm done in 101 microseconds.
done (control).
New request is data TestPerson51
Reading next request from channel (control) ...reply to request 51:3
Process Comm done in 102 microseconds.
done (control).
New request is data TestPerson52
Reading next request from channel (control) ...reply to request 52:62
Process Comm done in 107 microseconds.
done (control).
New request is data TestPerson53
Reading next request from channel (control) ...reply to request 53:18
Process Comm done in 100 microseconds.
done (control).
New request is data TestPerson54
Reading next request from channel (control) ...reply to request 54:22
Process Comm done in 100 microseconds.
done (control).
New request is data TestPerson55
Reading next request from channel (control) ...reply to request 55:27
Process Comm done in 102 microseconds.
done (control).
New request is data TestPerson56
Reading next request from channel (control) ...reply to request 56:57
Process Comm done in 100 microseconds.
done (control).
New request is data TestPerson57
Reading next request from channel (control) ...reply to request 57:45
```

Sample Output of Function_Time_Delay.C:



```
sun.cse.tamu.edu - PuTTY

Function Comm done in 11 microseconds.

Request data TestPerson92
Reply to request 29

Function Comm done in 11 microseconds.

Request data TestPerson93
Reply to request 44

Function Comm done in 11 microseconds.

Request data TestPerson94
Reply to request 75

Function Comm done in 11 microseconds.

Request data TestPerson95
Reply to request 93

Function Comm done in 11 microseconds.

Request data TestPerson96
Reply to request 14

Function Comm done in 12 microseconds.

Request data TestPerson97
Reply to request 39

Function Comm done in 11 microseconds.

Request data TestPerson98
Reply to request 88

Function Comm done in 11 microseconds.

Request data TestPerson99
Reply to request 21

Function Comm done in 11 microseconds.

[zekorov]@sun ~/CSCE_313_504/MP2_Source> (19:03:14 10/05/13)
:: █
```