

Analysis of PEX4b

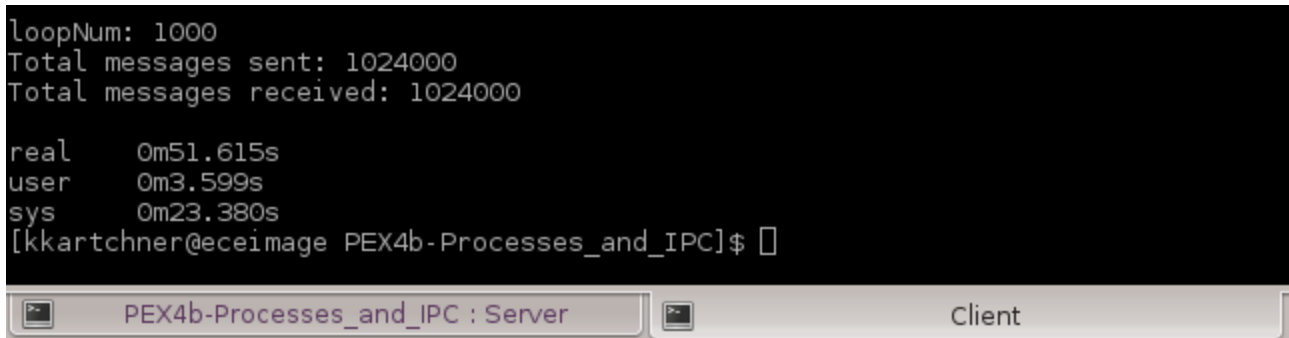
By: Ky Kartchner

As seen in the tables below, the system time for Phase 2 is unexpectedly slower than Phase 1 for every value of loopNum. This is not as expected because in Phase 1, the client process has to wait for an answer back from the server process before sending another problem. But in Phase 2, it does not. It is able to send problems to the server as fast as possible without any waiting. So, in theory, Phase 2 should be faster in this case just as Phase 2 in PEX4a. It would take exploration to figure out why, but it could be that compile optimization was not on or that using a forked process in this case simply is not faster.

Phase 1 – Client:

```
loopNum: 1000
Total messages sent: 1024000
Total messages received: 1024000

real    0m51.615s
user    0m3.599s
sys     0m23.380s
[kkartchner@eceimage PEX4b-Processes_and_IPC]$
```

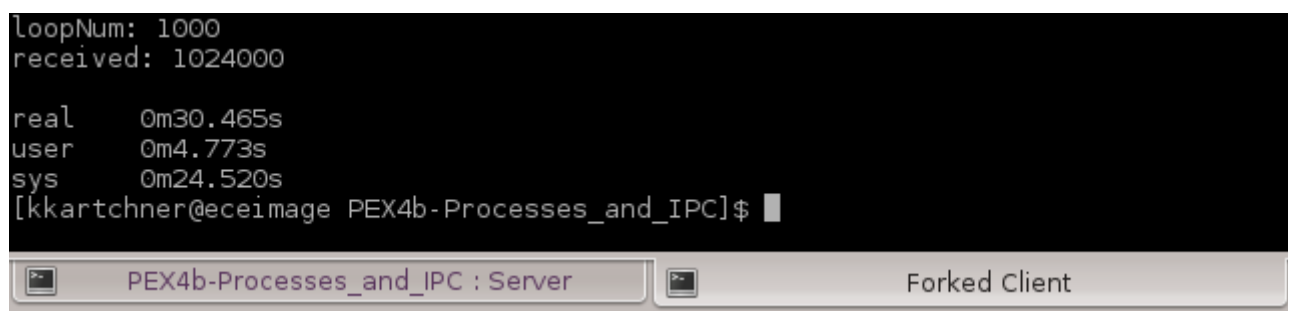
A terminal window titled "PEX4b-Processes_and_IPC : Server" with a "Client" tab. The terminal output shows the results of a client process for loopNum 1000. It reports 1,024,000 messages sent and received. Timing statistics show a real time of 51.615s, user time of 3.599s, and system time of 23.380s. The prompt is [kkartchner@eceimage PEX4b-Processes_and_IPC]\$.

LoopNum	System Time
1	0m0.037s
10	0m0.243s
100	0m2.355s
1000	0m23.380s

Phase 2 – Forked Client:

```
loopNum: 1000
received: 1024000

real    0m30.465s
user    0m4.773s
sys     0m24.520s
[kkartchner@eceimage PEX4b-Processes_and_IPC]$
```

A terminal window titled "PEX4b-Processes_and_IPC : Server" with a "Forked Client" tab. The terminal output shows the results of a forked client process for loopNum 1000. It reports 1,024,000 messages received. Timing statistics show a real time of 30.465s, user time of 4.773s, and system time of 24.520s. The prompt is [kkartchner@eceimage PEX4b-Processes_and_IPC]\$.

LoopNum	System Time
1	0m0.038s
10	0m0.264s
100	0m2.526s
1000	0m24.520s