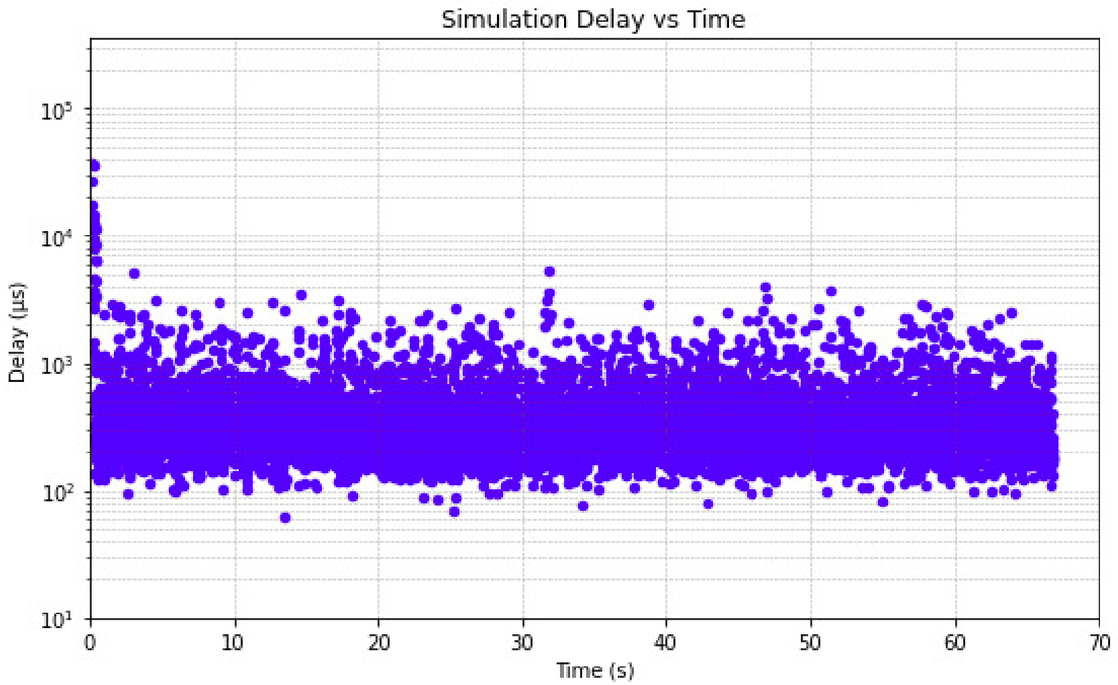
Testes de latência

Resultados anteriores

	Mínimo	Média	Máximo
Socket	**13**	**390**	**71273**
SharedMem	**1**	**207**	**68452**



(Gráfico está em escala logarítmica)

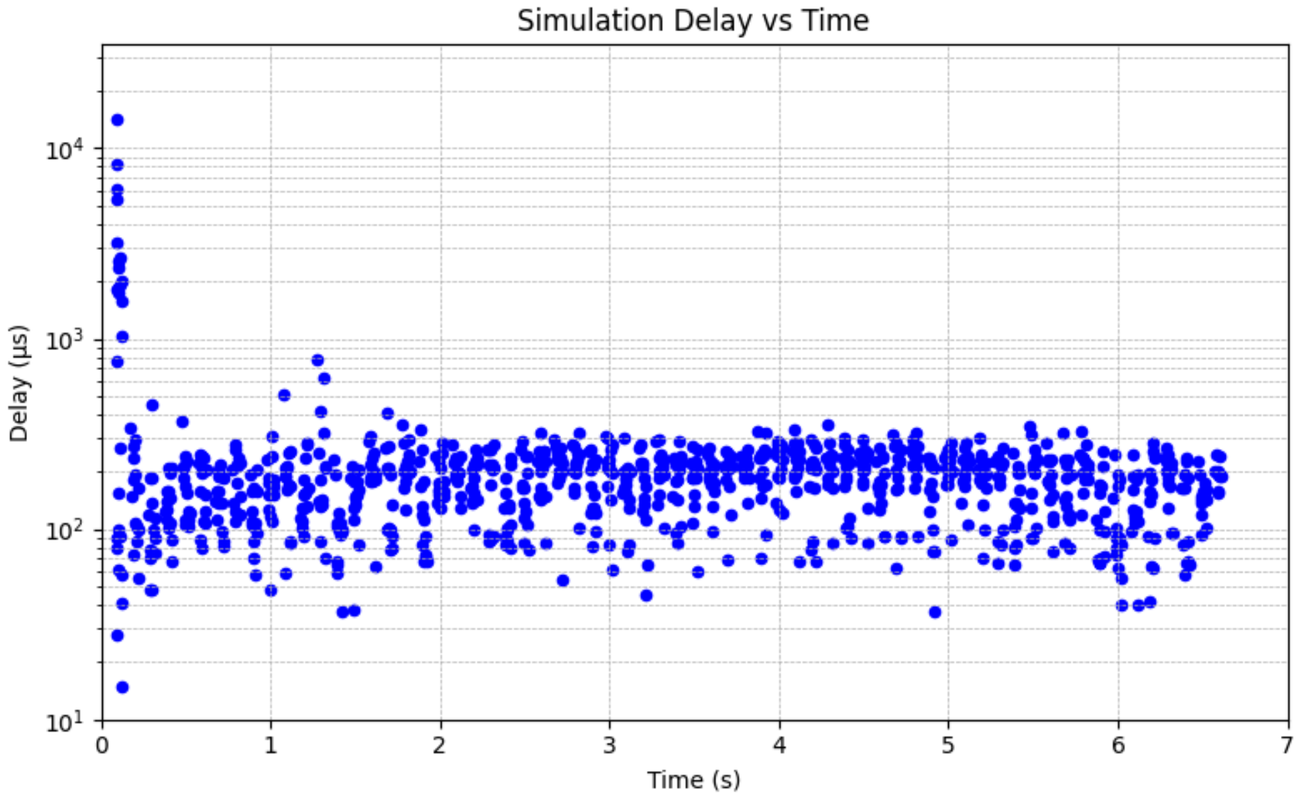
Processo de Debug

- Verificar se havia um erro lógico no código do teste
- Eliminar camadas da API a fim de identificar aonde estava o atraso
- Teste apenas com Engine de sockets
- Teste apenas com sockets
- Flag O_ASYNC e inicialização do programa

Processo de Debug

- **Verificar se havia um erro lógico no código do teste**
- Eliminar camadas da API a fim de identificar aonde estava o atraso
- Teste apenas com Engine de sockets
- Teste apenas com sockets
- Flag O_ASYNC e inicialização do programa

- Máximo de 14173 micro
- Altos delays no inicio do teste continuam

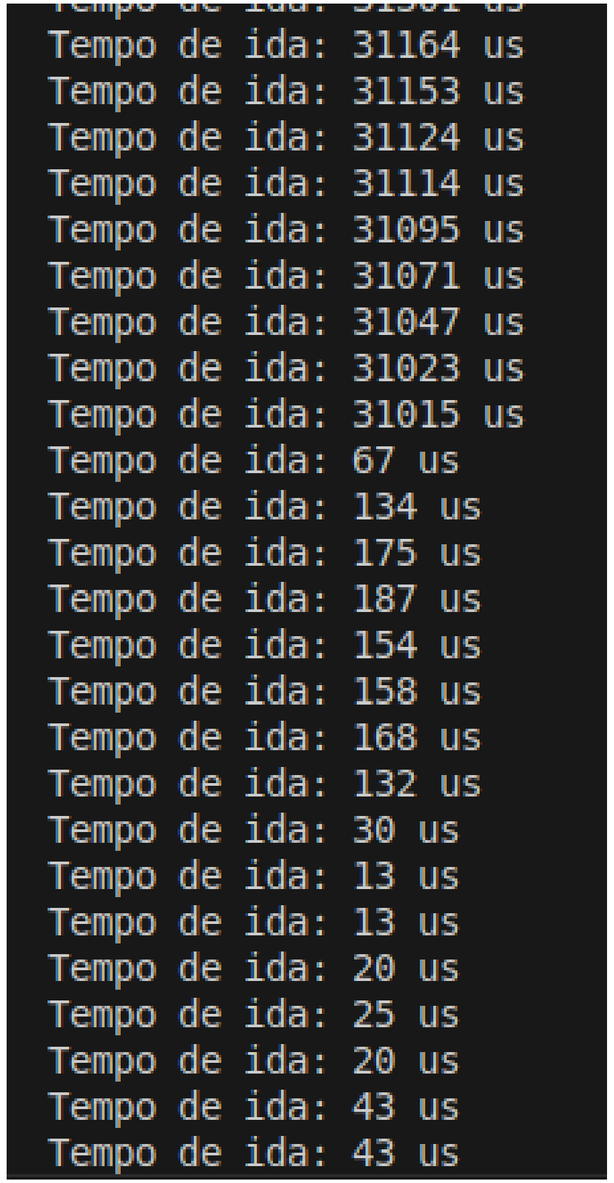


Processo de Debug

- Verificar se havia um erro lógico no código do teste
- **Eliminar camadas da API a fim de identificar aonde estava o atraso**
- Teste apenas com Engine de sockets
- Teste apenas com sockets
- Flag O_ASYNC e inicialização do programa

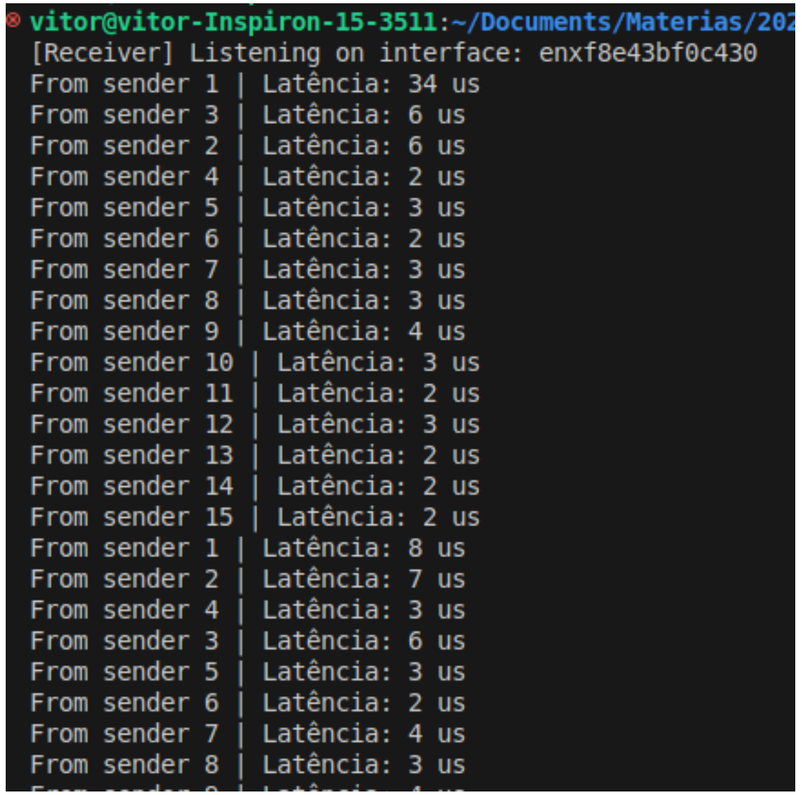
Processo de Debug

- Verificar se havia um erro lógico no código do teste
- Eliminar camadas da API a fim de identificar aonde estava o atraso
- **Teste apenas com Engine de sockets**
- Teste apenas com sockets
- Flag O_ASYNC e inicialização do programa



Processo de Debug

- Verificar se havia um erro lógico no código do teste
- Eliminar camadas da API a fim de identificar aonde estava o atraso
- Teste apenas com Engine de sockets
- **Teste apenas com sockets**
- Flag O_ASYNC e inicialização do programa

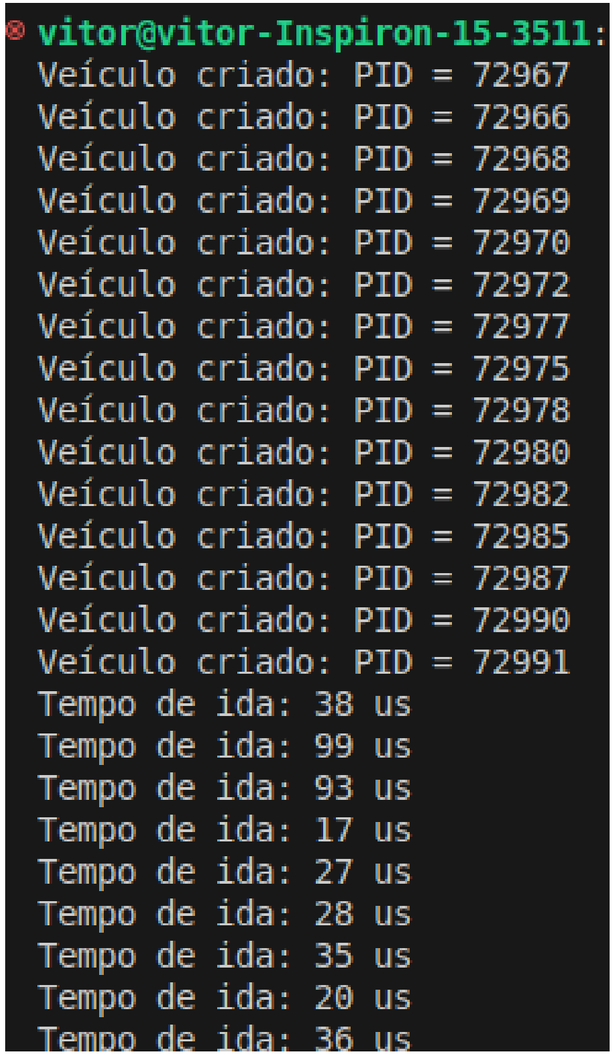


Processo de Debug

- Verificar se havia um erro lógico no código do teste
- Eliminar camadas da API a fim de identificar aonde estava o atraso
- Teste apenas com Engine de sockets
- Teste apenas com sockets
- **Flag O_ASYNC e inicialização do programa**

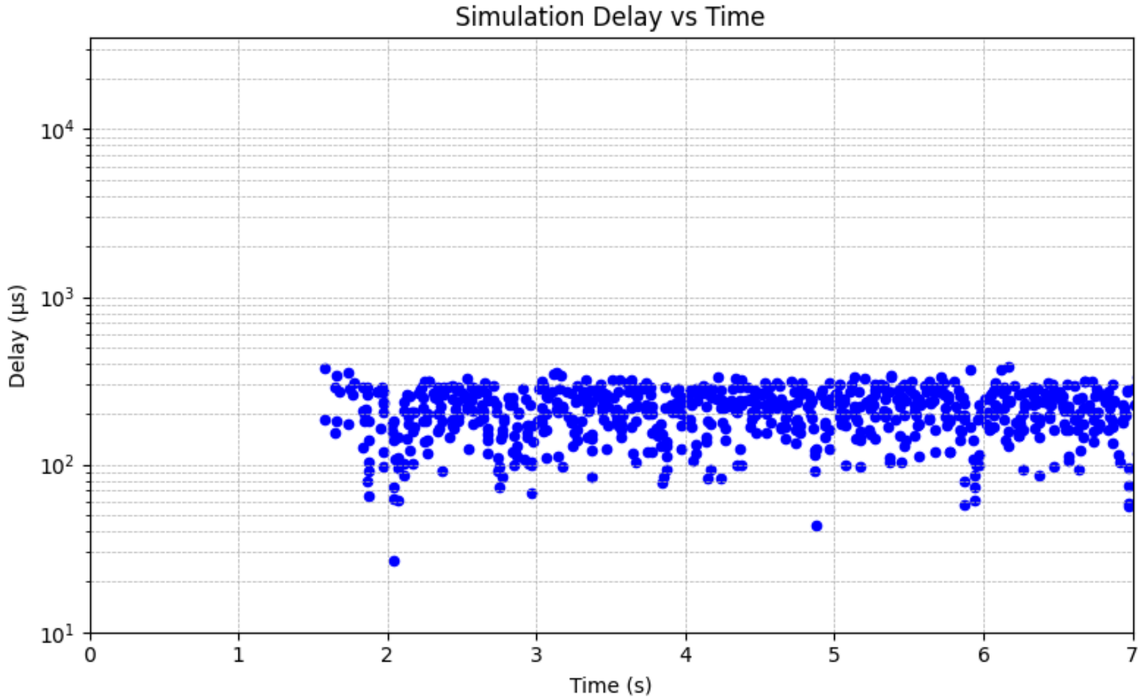
```
132         // O_ASYNC faz com que o socket levante o sinal SIGIO quando operacoes de
133         // I/O acontecerem
134         // O_NONBLOCK faz com que operações normalmente bloqueantes
135         // não bloqueiem
136         if (fcntl(sockfd, F_SETFL, flags | O_ASYNC | O_NONBLOCK) <
137             0) {
138         perror("fcntl F_SETFL");
139         exit(EXIT_FAILURE);
140         }
```

- Ao remover O_ASYNC, teste apenas com a Engine passou a dar resultados normais no início do teste
- Porém, remover a flag não é uma solução, sem ela não é possível receber SIGIOs



Resultados

	Mínimo	Média	Máximo
Socket	**16**	**211.93**	**721**
SharedMem	**1**	**1.53**	**58**



Processo de Debug parte 2

- Testar recepção assíncrona apenas com sockets
- Testar recepção assíncrona a partir da Engine
- Testar recepção assíncrona a partir da NIC

Processo de Debug parte 2

- **Testar recepção assíncrona apenas com sockets**
- Testar recepção assíncrona a partir da Engine
- Testar recepção assíncrona a partir da NIC

Tempos de delay entre envio e recebimento:
2, 244, 208, 118, 91, 89, 44, 40, 21, 93, 33, 25, 82, 35, 20, 90, 108, 78, 81, 27, 27, 36, 44, 29, 95, 36, 19, 85, 16, 22, 87, 98, 86, 45, 19, 22, 21, 41, 14, 96, 36, 18, 99, 32, 18, 1
00, 71, 42, 12, 29, 17, 45, 44, 26, 15, 20, 16, 21, 18, 16, 26, 109, 113, 55, 40, 44, 23, 25, 16, 41, 18, 17, 21, 18, 17, 103, 189, 59, 61, 50, 28, 44, 42, 34, 18, 16, 23, 22, 35, 18,
89, 29, 13, 11, 9, 10, 6, 8, 14, 12, 10, 8, 7, 6, 6, 49, 86, 63, 65, 29, 8, 25, 28, 17, 13, 12, 18, 17, 29, 20, 19, 97, 49, 28, 15, 18, 22, 30, 11, 18, 16, 24, 21, 37, 19, 22, 71, 13,
29, 25, 27, 12, 16, 29, 16, 19, 13, 17, 18, 14, 13, 87, 43, 46, 30, 64, 53, 46, 26, 25, 16, 24, 20, 30, 18, 18, 29, 11, 10, 8, 8, 9, 14, 6, 6, 12, 7, 9, 9, 7, 6, 29, 11, 10, 8, 6, 10,
15, 6, 6, 16, 14, 7, 10, 7, 6, 75, 38, 36, 9, 15, 24, 23, 15, 14, 36, 16, 23, 14, 14, 14, 90, 56, 39, 21, 17, 27, 27, 17, 16, 38, 20, 26, 15, 17, 15, 44, 9, 9, 15, 9, 14, 10, 19, 8, 12
, 9, 8, 8, 10, 8, 72, 47, 35, 30, 15, 25, 28, 28, 22, 16, 15, 17, 14, 15, 14, 90, 39, 47, 41, 17, 31, 25, 16, 28, 16, 16, 15, 14, 16, 14, 86, 52, 32, 35, 20, 29, 25, 18, 19, 16, 16, 15
, 14, 15, 15, 74, 46, 34, 38, 17, 27, 26, 18, 17, 14, 15, 16, 14, 15, 15, 40, 17, 11, 18, 8, 15, 13, 9, 10, 9, 8, 9, 8, 9, 8, 117, 106, 109, 41, 47, 40, 42, 15, 29, 88, 44, 20, 86, 48,
23, 79, 52, 23, 39, 85, 42, 23, 23, 18, 18, 25, 19, 29, 19, 34, 35, 15, 8, 9, 12, 11, 6, 10, 10, 16, 8, 8, 10, 9, 12, 33, 14, 7, 9, 6, 14, 6, 8, 6, 18, 18, 11, 9, 8, 9, 35, 13, 7, 9,
10, 12, 9, 7, 7, 8, 10, 7, 18, 15, 9, 35, 9, 18, 8, 6, 12, 10, 10, 7, 8, 4, 7, 10, 9, 7, 40, 13, 25, 11, 9, 12, 7, 6, 7, 16, 14, 10, 13, 17, 11, 51, 23, 16, 8, 10, 10, 15, 14, 13, 10,
7, 6, 6, 6, 9, 30, 8, 12, 7, 7, 10, 14, 8, 8, 16, 15, 8, 6, 7, 9, 52, 23, 12, 7, 7, 10, 13, 6, 13, 6, 10, 6, 6, 9, 7, 80, 55, 32, 9, 13, 23, 41, 13, 13, 12, 13, 12, 11, 20, 14, 220, 22
2, 157, 160, 58, 55, 56, 63, 46, 29, 41, 18, 16, 106, 76, 82, 72, 40, 21, 28, 17, 16, 18, 26, 16, 35, 16, 14, 20, 15, 81, 57, 40, 12, 30, 49, 49, 39, 24, 15, 15, 14, 14, 20, 15, 87, 62
, 42, 12, 32, 42, 15, 28, 31, 17, 16, 15, 16, 22, 17, 87, 63, 36, 29, 29, 48, 18, 29, 31, 16, 16, 16, 15, 24, 16, 61, 26, 34, 18, 26, 31, 19, 19, 20, 21, 19, 15, 14, 15, 18, 161, 126,
61, 44, 33, 36, 9, 12, 16, 18, 19, 15, 14, 15, 23, 95, 64, 44, 37, 29, 39, 14, 13, 32, 16, 21, 15, 15, 16, 28, 74, 39, 20, 34, 27, 34, 9, 18, 23, 22, 15, 15, 14, 15, 25, 80, 53, 16, 21
, 27, 35, 10, 28, 10, 15, 19, 15, 14, 14, 22, 210, 213, 159, 161, 73, 56, 51, 45, 30, 25, 38, 17, 15, 30, 18, 187, 160, 50, 47, 48, 50, 26, 28, 23, 17, 60, 12, 18, 18, 80, 106, 82, 42,
25, 32, 24, 25, 25, 20, 27, 43, 18, 15, 16, 77, 195, 203, 142, 144, 77, 51, 57, 37, 17, 20, 42, 17, 16, 18, 24, 190, 192, 125, 128, 62, 36, 45, 15, 31, 18, 40, 17, 17, 18, 93, 90, 60,
38, 15, 32, 19, 16, 30, 25, 17, 43, 18, 16, 17, 23, 153, 130, 51, 35, 24, 15, 15, 23, 15, 13, 31, 13, 12, 13, 17, 182, 189, 125, 127, 72, 46, 47, 34, 11, 18, 36, 16, 17, 18, 27, 199,
202, 139, 142, 80, 56, 61, 50, 14, 23, 40, 18, 16, 34, 30, 171, 174, 118, 121, 60, 43, 35, 32, 22, 16, 35, 16, 14, 15, 25, 221, 224, 119, 121, 89, 65, 62, 40, 26, 27, 39, 18, 16, 17, 7
6, 206, 209, 142, 145, 90, 55, 64, 52, 23, 22, 40, 17, 16, 18, 27, 178, 185, 121, 123, 60, 50, 32, 30, 26, 19, 39, 19, 16, 33, 27, 148, 156, 131, 133, 62, 51, 35, 33, 24, 29, 40, 18, 1
7, 17, 96, 27, 7, 11, 8, 11, 11, 6, 7, 8, 6, 12, 7, 6, 6, 8, 183, 185, 113, 115, 60, 52, 33, 27, 25, 15, 34, 15, 13, 15, 25, 185, 186, 105, 105, 55, 25, 29, 21, 21, 11, 24, 11, 10, 12,
14, 185, 188, 126, 128, 62, 56, 36, 32, 22, 18, 36, 17, 15, 16, 80, 197, 200, 132, 134, 77, 50, 55, 46, 10, 18, 39, 18, 16, 18, 27, 194, 197, 146, 150, 85, 56, 56, 39, 22, 58, 28, 23,
19, 23, 22, 207, 209, 145, 146, 95, 60, 59, 66, 56, 21, 43, 18, 16, 24, 29, 93, 43, 34, 38, 30, 23, 16, 18, 21, 16, 43, 18, 16, 17, 22, 174, 182, 188, 123, 124, 62, 47, 36, 24, 24, 37
, 17, 15, 17, 81, 167, 169, 164, 100, 103, 56, 54, 28, 27, 29, 43, 19, 17, 19, 105, 169, 175, 177, 119, 110, 67, 38, 43, 28, 20, 43, 18, 17, 35, 26, 176, 187, 155, 133, 104, 53, 50, 27
, 25, 20, 42, 18, 16, 19, 78, 121, 122, 123, 75, 58, 34, 21, 3, 14, 11, 19, 9, 8, 9, 45, 169, 172, 174, 116, 101, 99, 40, 31, 18, 15, 30, 13, 12, 13, 67, 184, 187, 180, 138, 125, 73, 7
0, 56, 33, 17, 41, 18, 17, 19, 84, 171, 174, 169, 105, 97, 58, 50, 31, 24, 16, 39, 18, 14, 16, 22, 176, 151, 71, 49, 41, 33, 18, 23, 27, 17, 41, 18, 16, 18, 26, 141, 144, 145, 136, 63,
53, 51, 40, 20, 24, 41, 18, 17, 20, 97, 89, 73, 62, 38, 36, 32, 18, 24, 18, 17, 41, 17, 16, 18, 84, 202, 205, 198, 173, 148, 101, 72, 56, 34, 27, 50, 18, 16, 18, 92, 163, 165, 152, 11
5, 124, 60, 52, 23, 25, 22, 43, 19, 16, 34, 105, 50, 20, 22, 23, 27, 24, 18, 13, 17, 12, 57, 11, 64, 26, 77, 142, 120, 49, 21, 34, 23, 14, 12, 38, 12, 21, 23, 15, 66, 13, 54, 24, 24, 2
3, 28, 15, 15, 25, 15, 15, 29, 12, 14, 15, 14, 195, 198, 180, 182, 81, 51, 56, 20, 32, 93, 34, 18, 85, 39, 22, 50, 20, 21, 22, 28, 88, 70, 40, 16, 68, 20, 79, 22, 16, 14, 176, 178, 174
, 176, 66, 54, 40, 29, 7, 20, 24, 17, 16, 20, 16, 149, 161, 157, 160, 66, 55, 45, 27, 9, 86, 27, 24, 86, 45, 18, 162, 165, 161, 164, 69, 49, 41, 31, 7, 88, 32, 23, 119, 54, 29, 31, 11,
9, 8, 11, 10, 8, 8, 10, 8, 8, 6, 6, 8, 6, 31, 10, 10, 11, 12, 15, 12, 12, 10, 10, 9, 6, 6, 8, 6, 38, 15, 11, 8, 14, 22, 13, 9, 14, 12, 13, 8, 14, 10, 15, 29, 13, 19, 18, 9, 8, 10, 8,
7, 8, 11, 6, 6, 6, 7, 71, 41, 35, 14, 16, 24, 26, 18, 16, 21, 32, 19, 16, 27, 16, 56, 23, 22, 13, 11, 18, 27, 14, 16, 21, 22, 13, 14, 12, 13, 41, 10, 11, 8, 15, 8, 7, 7, 7, 10, 17, 9,
10, 8, 8, 47, 48, 35, 13, 9, 23, 16, 16, 14, 11, 15, 16, 11, 9, 8, 36, 14, 16, 10, 8, 12, 16, 19, 12, 8, 9, 6, 8, 9, 8, 3248, 3206, 3195, 3185, 3049, 3049, 3030, 3018, 2929, 2903, 2058
, 2041, 2030, 1043, 59, 65, 28, 27, 16, 13, 15, 14, 23, 21, 14, 14, 14, 29, 14, 18, 119, 91, 48, 19, 7, 9, 6, 9, 16, 7, 9, 7, 10, 7, 6, 65, 34, 26, 15, 14, 14, 14, 18, 20, 15, 20, 14,
25, 15, 18, 83, 18, 23, 15, 15, 15, 18, 24, 23, 15, 19, 20, 25, 15, 18, 80, 47, 25, 17, 13, 14, 15, 20, 22, 14, 18, 13, 24, 14,

Processo de Debug parte 2

- Testar recepção assíncrona apenas com sockets
- **Testar recepção assíncrona a partir da Engine**
  - Resultados idênticos ao anterior
- Testar recepção assíncrona a partir da NIC

Processo de Debug parte 2

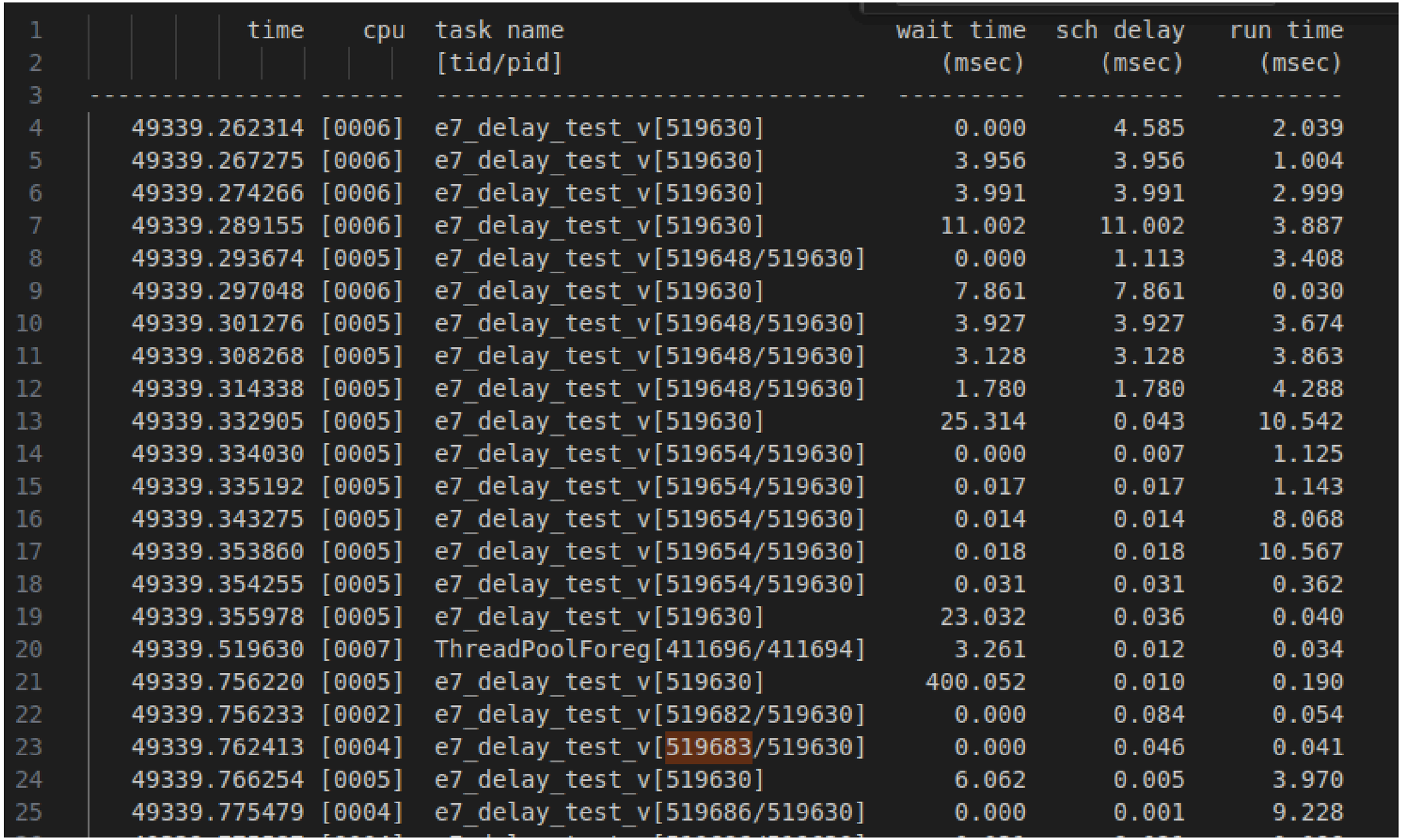
- Testar recepção assíncrona apenas com sockets
- Testar recepção assíncrona a partir da Engine
- **Testar recepção assíncrona a partir da NIC**
  - Foram encontrados atrasos de 1 a 2 milissegundos devido ao construtor da NIC inicializar as pools de buffer.

Ferramentas para debugging

- **Strace**: ferramenta utilizada para verificar quanto tempo o código demorou para retornar de uma syscall
  - Os tempos para sair de um recvfrom raramente passavam de 400 microssegundos utilizando a ferramenta, testando sem ela, apenas pelo código, era ainda menor.

Ferramentas para debugging

- **Perf**: ferramenta utilizada para verificar quanto tempo se passou desde que o processo/thread foi acordado até o momento em que ele realmente começou a executar.
  - Realmente, no início do programa os processos demoravam milissegundos para executar.



time	cpu	task name [tid/pid]	wait time (msec)	sch delay (msec)	run time (msec)
49339.262314	[0006]	e7_delay_test_v[519630]	0.000	4.585	2.039
49339.267275	[0006]	e7_delay_test_v[519630]	3.956	3.956	1.004
49339.274266	[0006]	e7_delay_test_v[519630]	3.991	3.991	2.999
49339.289155	[0006]	e7_delay_test_v[519630]	11.002	11.002	3.887
49339.293674	[0005]	e7_delay_test_v[519648/519630]	0.000	1.113	3.408
49339.297048	[0006]	e7_delay_test_v[519630]	7.861	7.861	0.030
49339.301276	[0005]	e7_delay_test_v[519648/519630]	3.927	3.927	3.674
49339.308268	[0005]	e7_delay_test_v[519648/519630]	3.128	3.128	3.863
49339.314338	[0005]	e7_delay_test_v[519648/519630]	1.780	1.780	4.288
49339.332905	[0005]	e7_delay_test_v[519630]	25.314	0.043	10.542
49339.334030	[0005]	e7_delay_test_v[519654/519630]	0.000	0.007	1.125
49339.335192	[0005]	e7_delay_test_v[519654/519630]	0.017	0.017	1.143
49339.343275	[0005]	e7_delay_test_v[519654/519630]	0.014	0.014	8.068
49339.353860	[0005]	e7_delay_test_v[519654/519630]	0.018	0.018	10.567
49339.354255	[0005]	e7_delay_test_v[519654/519630]	0.031	0.031	0.362
49339.355978	[0005]	e7_delay_test_v[519630]	23.032	0.036	0.040
49339.519630	[0007]	ThreadPoolForeg[411696/411694]	3.261	0.012	0.034
49339.756220	[0005]	e7_delay_test_v[519630]	400.052	0.010	0.190
49339.756233	[0002]	e7_delay_test_v[519682/519630]	0.000	0.084	0.054
49339.762413	[0004]	e7_delay_test_v[519683/519630]	0.000	0.046	0.041
49339.766254	[0005]	e7_delay_test_v[519630]	6.062	0.005	3.970
49339.775479	[0004]	e7_delay_test_v[519686/519630]	0.000	0.001	9.228

51930 é o PID do processo, 519683 é o TID da thread de recebimento daquele processo

Outra fonte de delay na API...

- Testei quanto tempo demorava para descer e subir a pilha
- Executar o calculo do MAC pela primeira vez causava um delay de pelo menos 1 milissegundo. O calculo é feito utilizando openssl