Crimson Seastore: single vs. dual Seastar reactor per core – progress report

José Juan Palacios Pérez
Ceph IBM,
Manchester UK

May 20, 2025

Abstract

In this brief report we summarise the performance results for the comparison between the Crimson SeaStore using the following configurations:

- single Seastar reactor per physical core: up to 28 reactors, single NUMA socket,
- dual Seastar reactor per physical core: up to 56 reactors, single NUMA socket.
- We used the same ceph dev build from main branch (hash 6aab5c07ae) for both.
- We only used the balanced OSD algorithm.

Our preliminary conclusions:

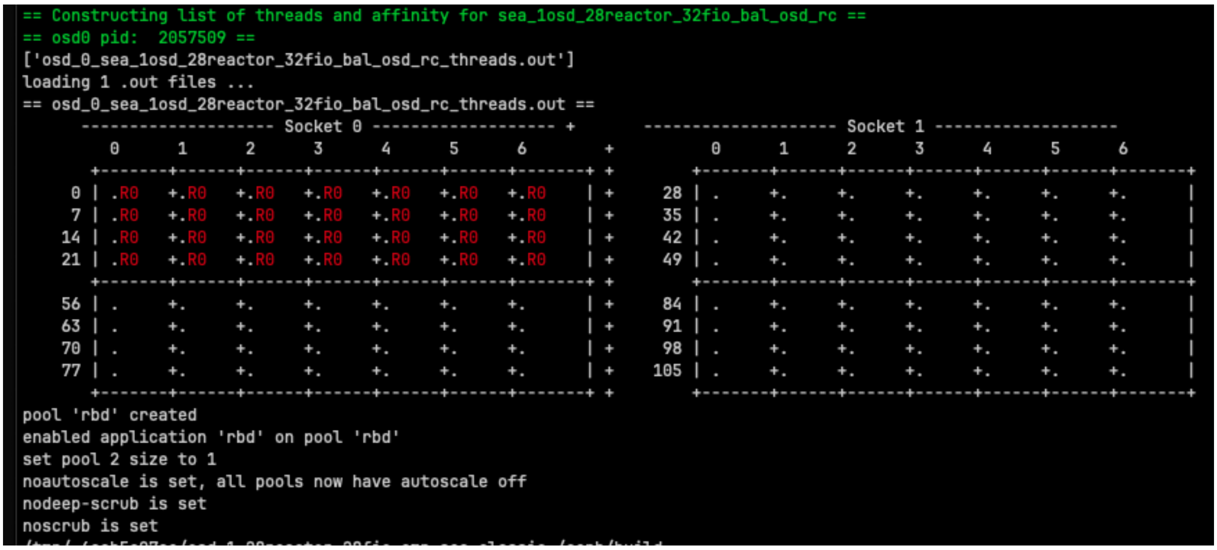
- The performance of the Crimson SeaStore using the dual Seastar reactor per physical core achieves better performance across the four typical workloads. This suggest that Seastar is HT-friendly in the sense that reactors don't seem to interfere each other use of the physical CPU.

Contents

1. Seastore OSD: single vs. dual Seastar reactor per core

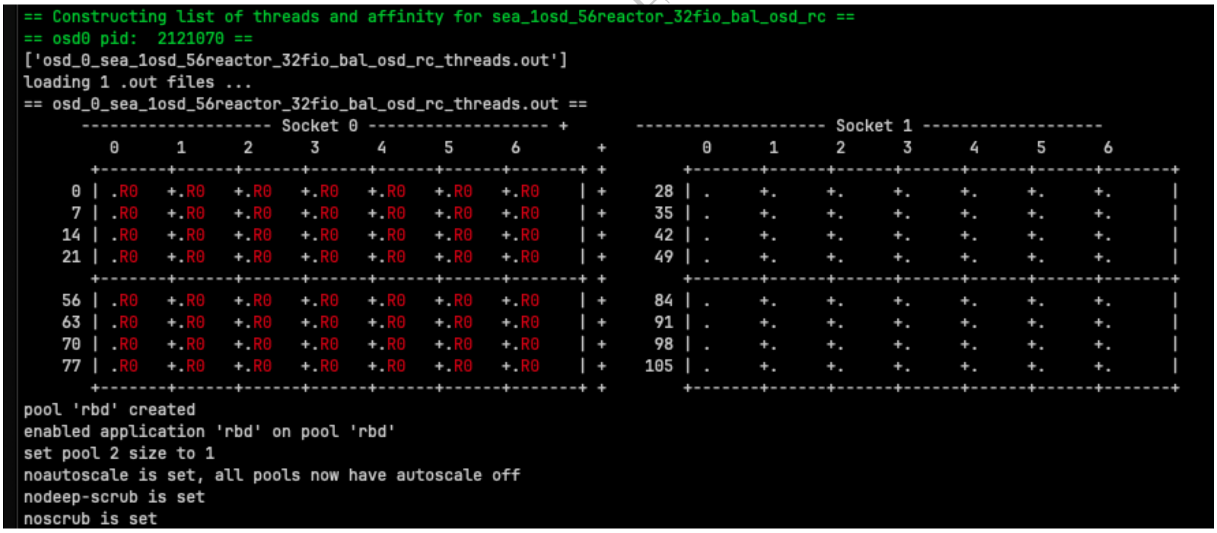
In this Chapter we show the comparison of Seastore (build 6aab5c07ae) on longer duration test sets, producing response latency curves.

The single reactor per CPU core configuration is shown in Figure 1.1, whereas the dual reactor configuration is shown in Figure 1.2. In both cases, we use the balanced OSD method.



```
== Constructing list of threads and affinity for sea_1osd_28reactor_32fio_bal_osd_rc ==
== osd0 pid:  2057509 ==
['osd_0_sea_1osd_28reactor_32fio_bal_osd_rc_threads.out']
loading 1 .out files ...
== osd_0_sea_1osd_28reactor_32fio_bal_osd_rc_threads.out ==
        ------------------- Socket 0 ------------------- +       ------------------- Socket 1 ------------------
           0      1      2      3      4      5      6        +         0      1      2      3      4      5      6
         +------+------+------+------+------+------+-------+ +       +------+------+------+------+------+------+-------+
       0 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +    28 | .     +.     +.     +.     +.     +.     +.      |
       7 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +    35 | .     +.     +.     +.     +.     +.     +.      |
      14 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +    42 | .     +.     +.     +.     +.     +.     +.      |
      21 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +    49 | .     +.     +.     +.     +.     +.     +.      |
         +------+------+------+------+------+------+-------+ +       +------+------+------+------+------+------+-------+
      56 | .     +.     +.     +.     +.     +.     +.      | +    84 | .     +.     +.     +.     +.     +.     +.      |
      63 | .     +.     +.     +.     +.     +.     +.      | +    91 | .     +.     +.     +.     +.     +.     +.      |
      70 | .     +.     +.     +.     +.     +.     +.      | +    98 | .     +.     +.     +.     +.     +.     +.      |
      77 | .     +.     +.     +.     +.     +.     +.      | +   105 | .     +.     +.     +.     +.     +.     +.      |
         +------+------+------+------+------+------+-------+ +       +------+------+------+------+------+------+-------+
pool 'rbd' created
enabled application 'rbd' on pool 'rbd'
set pool 2 size to 1
noautoscale is set, all pools now have autoscale off
nodeep-scrub is set
noscrub is set
```

Figure 1.1: Single reactor per physical CPU core configuration.



```
== Constructing list of threads and affinity for sea_1osd_56reactor_32fio_bal_osd_rc ==
== osd0 pid:  2121070 ==
['osd_0_sea_1osd_56reactor_32fio_bal_osd_rc_threads.out']
loading 1 .out files ...
== osd_0_sea_1osd_56reactor_32fio_bal_osd_rc_threads.out ==
        ------------------- Socket 0 ------------------- +       ------------------- Socket 1 ------------------
           0      1      2      3      4      5      6        +         0      1      2      3      4      5      6
         +------+------+------+------+------+------+-------+ +       +------+------+------+------+------+------+-------+
       0 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +    28 | .     +.     +.     +.     +.     +.     +.      |
       7 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +    35 | .     +.     +.     +.     +.     +.     +.      |
      14 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +    42 | .     +.     +.     +.     +.     +.     +.      |
      21 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +    49 | .     +.     +.     +.     +.     +.     +.      |
         +------+------+------+------+------+------+-------+ +       +------+------+------+------+------+------+-------+
      56 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +    84 | .     +.     +.     +.     +.     +.     +.      |
      63 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +    91 | .     +.     +.     +.     +.     +.     +.      |
      70 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +    98 | .     +.     +.     +.     +.     +.     +.      |
      77 | .R0   +.R0   +.R0   +.R0   +.R0   +.R0   +.R0    | +   105 | .     +.     +.     +.     +.     +.     +.      |
         +------+------+------+------+------+------+-------+ +       +------+------+------+------+------+------+-------+
pool 'rbd' created
enabled application 'rbd' on pool 'rbd'
set pool 2 size to 1
noautoscale is set, all pools now have autoscale off
nodeep-scrub is set
noscrub is set
```

Figure 1.2: Dual reactor per physical CPU core configuration.

Unfortunately, there seemed to have occurred several failures, since the OSD pid being monitored does not appear the whole duration of the test in the performance data. This is reflected in zero valued OSD measurements. The test drive script does not expected the pid for the process OSD to change for the lifetime of the cluster. Consequently this also affects the benchmark data, as shown by the considerable fluctuations.

Despite this, the results below show that the dual configuration achieves better performance than the single reactor configuration.

1.1 Random read 4k

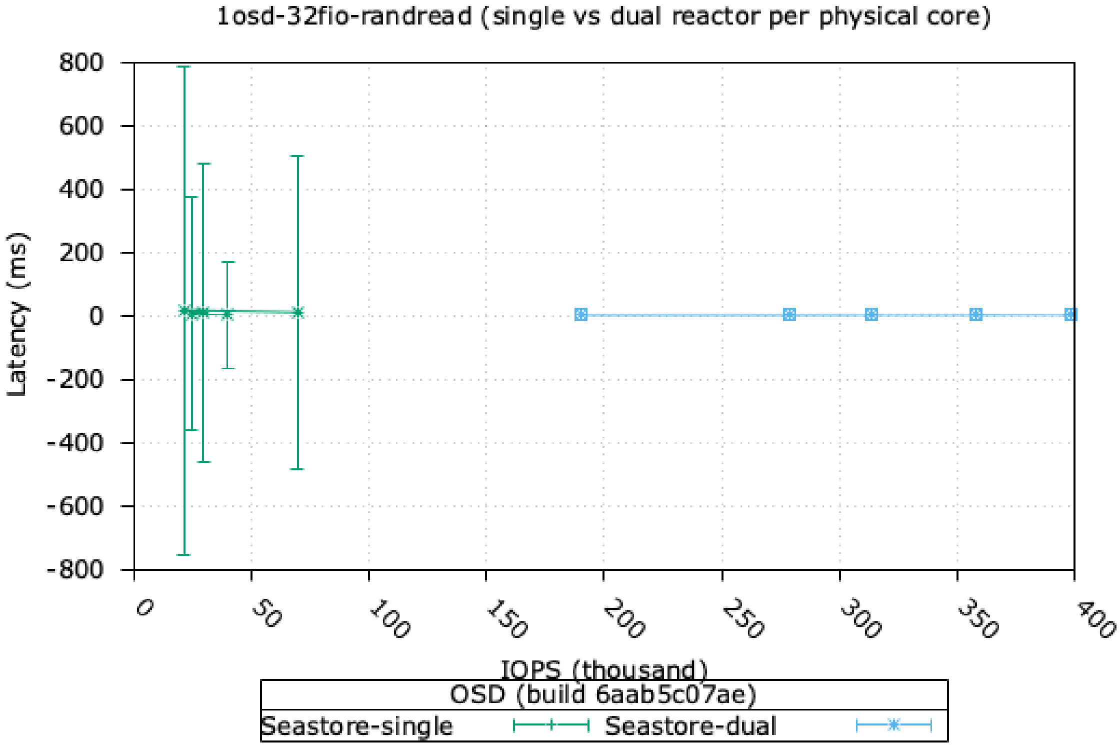


Figure 1.3: Response latency curves, IOPs vs. Latency

1.1.1 Single reactor

iodepth	iops	total_ios	clat_ms	clat_stdev	usr_cpu	sys_cpu	OSD_cpu	OSD_mem	FIO_cp
1	69601.46	3470816.00	0.45	74.71	1.05	0.85	850.03	632.78	247.00
2	40152.74	1399885.00	1.58	168.81	0.57	0.49	0.00	0.00	127.03
4	24441.49	852226.00	5.20	368.63	0.34	0.28	0.00	0.00	101.70
8	29791.14	1038668.00	8.52	472.16	0.40	0.30	0.00	0.00	109.96
16	21280.95	1185221.00	17.46	771.22	0.34	0.26	0.00	0.00	111.48
24	69517.65	2771321.00	10.94	494.88	0.82	0.61	0.00	0.00	197.18
32	162733.15	15462578.00	6.23	267.53	1.96	1.35	0.00	0.00	854.02
40	112628.71	7868242.00	11.25	493.07	1.28	0.88	0.00	0.00	434.93
52	34984.86	1219887.00	46.36	1097.90	0.40	0.28	0.00	0.00	110.94
64	139924.42	11175064.00	14.39	455.18	1.67	1.14	0.00	0.00	632.25

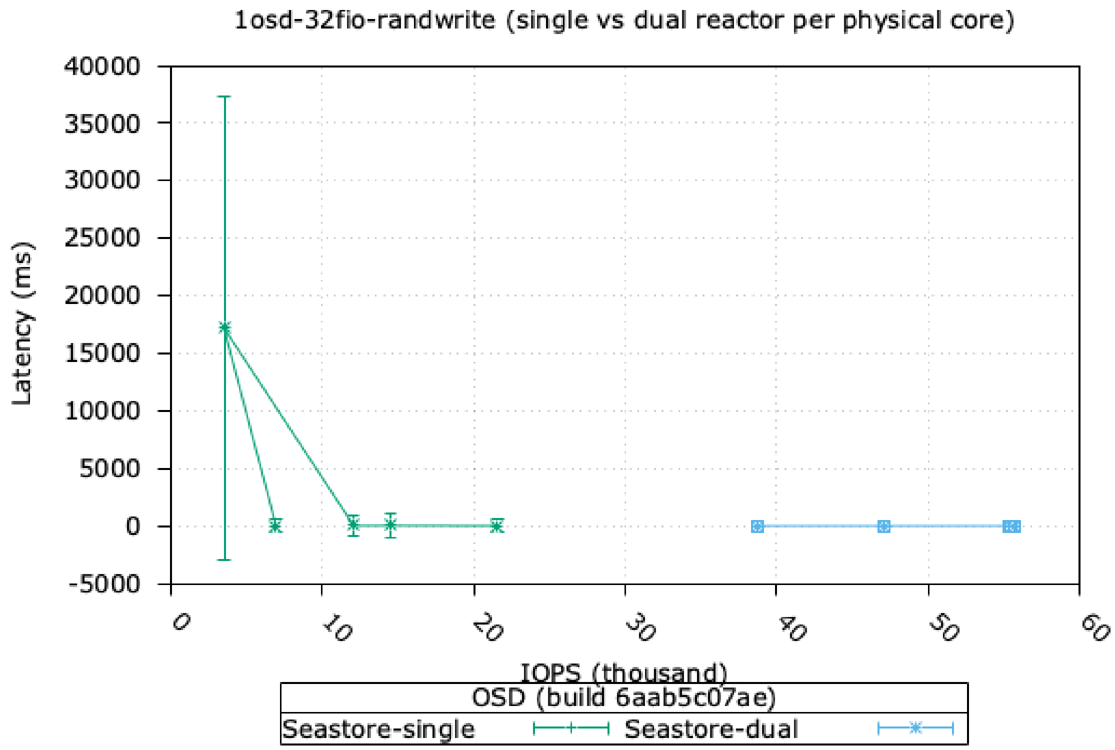
Table 1.1: Performance Throughput vs Latency vs CPU util: random read 4k single reactor per CPU core.

1.1.2 Dual reactor

iodepth	iops	total_ios	clat_ms	clat_stdev	usr_cpu	sys_cpu	OSD_cpu	OSD_mem	FIO_c
1	126347.94	37904636.00	0.25	0.06	1.94	1.66	4308.15	3491.53	763.1
2	190191.41	57057612.00	0.33	0.12	2.83	2.43	4776.95	3645.60	1183.
4	278786.24	83636152.00	0.45	0.19	4.21	3.37	5163.70	3645.60	1688.
8	313564.52	94069983.00	0.81	0.46	3.99	2.99	5251.90	3645.60	1619.
16	397880.95	119367467.00	1.28	1.91	5.39	3.60	5340.68	3645.60	2031.
24	357806.40	107347286.00	2.14	5.90	4.16	2.74	5349.39	3661.84	1584.
32	394166.02	118262025.00	2.58	12.86	5.23	3.48	5350.25	3662.40	1966.
40	336976.99	101105564.00	3.78	19.79	3.98	2.77	5327.89	3662.40	1537.
52	354364.60	106334185.00	4.67	26.15	4.93	3.39	5314.44	3662.40	1857.
64	323955.66	97205488.00	6.27	36.54	3.86	2.74	5311.75	3662.40	1507.

Table 1.2: Performance Throughput vs Latency vs CPU util: random read 4k dual reactor per CPU core.

1.2 Random write 4k



1.2.1 Single reactor

iodepth	iops	total_ios	clat_ms	clat_stdev	usr_cpu	sys_cpu	OSD_cpu	OSD_mem	FIO_cpu
1	462.66	20688.00	68.79	1718.76	0.01	0.01	0.00	0.00	2.35
2	6974.29	347787.00	9.12	542.11	0.14	0.10	0.00	0.00	39.02
4	3540.54	131.00	17200.65	20133.63	0.00	0.00	0.00	0.00	7.46
8	12089.40	723780.00	21.07	838.38	0.23	0.16	0.00	0.00	66.76
16	14523.08	1014684.00	34.72	1103.52	0.27	0.19	0.00	0.00	86.23
24	21568.72	2369475.00	10.10	508.13	1.34	0.96	0.00	0.00	177.59
32	8.41	419.00	35309.91	22885.21	0.00	0.00	0.00	0.00	7.28
40	191.32	9529.00	5953.88	16170.11	0.01	0.00	0.00	0.00	9.61
52	6594.02	1543.00	316.35	406.43	0.12	0.10	0.00	0.00	8.48
64	6062.91	2795.00	436.46	599.57	0.14	0.13	0.00	0.00	10.15

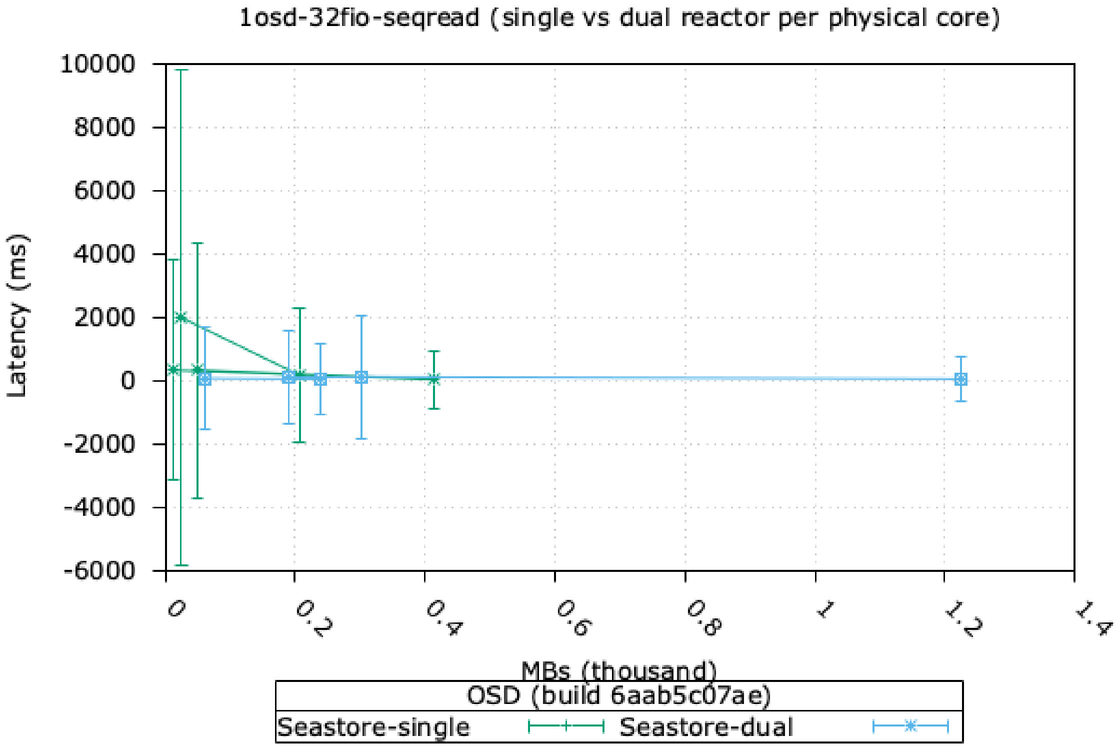
Table 1.3: Performance Throughput vs Latency vs CPU util: random write 4k, single reactor per CPU core.

1.2.2 Dual reactor

iodepth	iops	total_ios	clat_ms	clat_stdev	usr_cpu	sys_cpu	OSD_cpu	OSD_mem	FIO_cpu
1	26913.66	8074232.00	1.18	1.23	0.63	0.47	3494.35	3906.00	214.94
2	38782.02	11637243.00	1.64	2.02	0.77	0.58	4049.75	5192.88	287.14
4	47131.84	14139836.00	2.71	4.64	0.86	0.64	4662.55	5858.16	328.71
8	55309.01	16593310.00	4.62	6.81	0.99	0.70	4703.71	6204.80	359.32
16	55444.98	16637541.00	9.18	35.99	0.99	0.70	4824.54	6216.00	357.30
24	55646.51	16700073.00	13.66	66.18	1.00	0.70	4690.91	6216.00	352.63
32	56612.70	16988735.00	10.36	53.64	1.53	1.14	4687.60	6216.00	335.24
40	16345.45	6293.00	190.78	506.52	0.20	0.28	381.22	6216.00	24.09
52	15594.72	2410023.00	3.91	337.43	8.08	3.78	2217.48	6216.00	139.17
64	8685.92	2406.00	327.57	441.04	0.25	0.06	1.49	207.20	10.79

Table 1.4: Performance Throughput vs Latency vs CPU util: random write 4k dual reactor per CPU core.

1.3 Sequential read 64k



1.3.1 Single reactor

iodepth	bw	total_ios	clat_ms	clat_stdev	usr_cpu	sys_cpu	OSD_cpu	OSD_mem	FIO_cpu
1	176.76	123918.00	11.52	623.37	0.06	0.05	0.00	0.00	19.33
2	11.58	7183.00	313.32	3461.30	0.00	0.00	0.00	0.00	8.77
4	413.82	451651.00	19.72	901.86	0.12	0.10	0.00	0.00	64.49
8	49.72	42396.00	316.17	4025.28	0.03	0.03	0.00	0.00	13.67
16	205.82	127744.00	158.20	2136.51	0.06	0.05	0.00	0.00	28.51
24	24.80	12775.00	1971.73	7829.85	0.01	0.01	0.00	0.00	9.13
32	45.89	34797.00	1413.69	8086.52	0.02	0.02	0.00	0.00	15.40
40	775.16	852914.00	105.21	1883.88	0.18	0.16	0.00	0.00	110.62
52	1455.32	2063347.00	72.70	1246.43	0.32	0.30	0.00	0.00	237.54
64	1497.58	2109407.00	82.75	1307.90	0.34	0.32	0.00	0.00	241.70

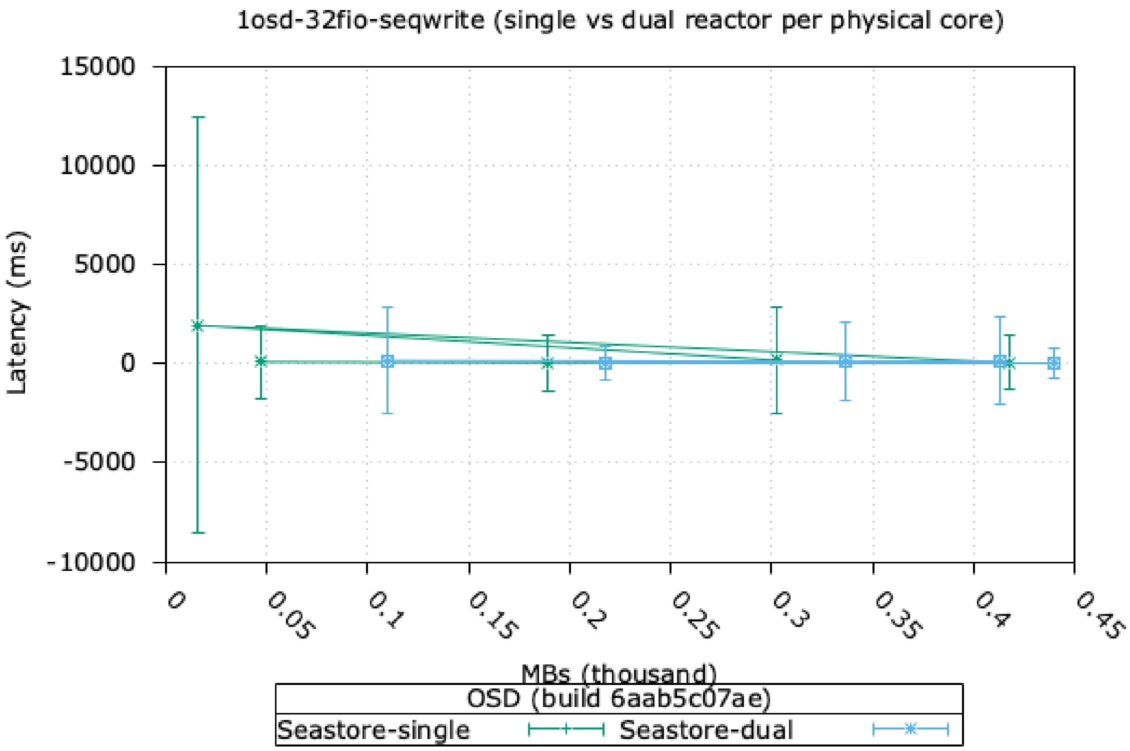
Table 1.5: Performance Throughput vs Latency vs CPU util: sequential read 64k, single reactor per CPU core.

1.3.2 Dual reactor

iodepth	bw	total_ios	clat_ms	clat_stdev	usr_cpu	sys_cpu	OSD_cpu	OSD_mem	FIO_cpu
1	327.03	561409.00	6.24	451.40	0.21	0.15	0.00	0.00	129.43
2	60.69	51958.00	56.80	1599.34	0.08	0.06	0.00	0.00	22.09
4	239.41	279987.00	34.10	1105.58	0.14	0.11	0.00	0.00	68.00
8	190.41	133305.00	85.60	1491.49	0.10	0.07	0.00	0.00	32.51
16	301.87	445172.00	104.02	1952.02	0.16	0.12	0.00	0.00	92.75
24	1224.36	2968089.00	40.03	713.62	0.36	0.30	0.00	0.00	294.68
32	564.11	706057.00	115.39	1627.78	0.20	0.16	0.00	0.00	127.31
40	840.67	1186983.00	96.60	1368.56	0.25	0.22	0.00	0.00	172.49
52	831.67	1110618.00	122.96	1575.43	0.24	0.21	0.00	0.00	158.24
64	552.20	569171.00	230.92	2305.86	0.18	0.15	0.00	0.00	92.94

Table 1.6: Performance Throughput vs Latency vs CPU util: sequential read 64k dual reactor per CPU core.

1.4 Sequential write 64k



1.4.1 Single reactor

iodepth	bw	total_ios	clat_ms	clat_stdev	usr_cpu	sys_cpu	OSD_cpu	OSD_mem	FIO_cpu
1	350.66	355427.00	5.80	447.95	0.28	0.10	0.00	0.00	50.17
2	47.33	36819.00	71.47	1841.48	0.06	0.02	0.00	0.00	15.10
4	189.52	177178.00	43.03	1418.21	0.12	0.05	0.00	0.00	31.24
8	417.77	488475.00	39.07	1347.09	0.29	0.09	0.00	0.00	67.74
16	16.33	14796.00	1922.68	10488.43	0.03	0.01	0.00	0.00	10.16
24	302.60	400513.00	161.87	2717.95	0.38	0.10	0.00	0.00	72.73
32	186.60	217308.00	343.57	4303.43	0.22	0.05	0.00	0.00	42.05
40	463.12	829952.00	176.29	2642.06	0.60	0.14	0.00	0.00	140.53
52	33.00	31818.00	3207.27	13813.57	0.05	0.01	0.00	0.00	11.36
64	599.70	1730285.00	170.90	2095.50	1.03	0.26	0.00	0.00	223.09

1.4.2 Dual reactor

iodepth	bw	total_ios	clat_ms	clat_stdev	usr_cpu	sys_cpu	OSD_cpu	OSD_mem	FIO_cpu
1	744.22	1510200.00	2.73	232.42	0.64	0.23	0.00	0.00	222.73
2	217.71	203618.00	18.72	833.16	0.22	0.09	0.00	0.00	42.53
4	439.65	582914.00	18.56	772.30	0.39	0.16	0.00	0.00	100.35
8	110.09	94153.00	145.91	2656.76	0.11	0.04	0.00	0.00	21.77
16	336.49	419443.00	97.01	1940.14	0.40	0.12	0.00	0.00	76.61
24	413.19	611754.00	118.51	2210.46	0.50	0.13	0.00	0.00	102.22
32	1135.64	5323596.00	57.63	49.16	1.45	0.36	0.00	0.00	234.83
40	1376.80	6454999.00	59.41	53.82	1.69	0.39	0.00	0.00	281.58
52	1470.67	6895065.00	72.28	69.71	1.84	0.41	0.00	0.00	309.51
64	1200.15	5626077.00	89.43	83.32	1.98	0.47	0.00	0.00	277.75

Table 1.7: Performance Throughput vs Latency vs CPU util: sequential write 64k, dual reactor per CPU core.