Joakim Frostegård, 2021-11-08

BitTorrent (UDP) tracker comparison

This is a performance comparison of several programs implementing the UDP BitTorrent tracker protocol made using aquatic_udp_load_test (https://github.com/greatest-ape/aquatic).

Setup

Tested trackers

Tracker	URL	Commit
aquatic_udp *	https://github.com/greatest-ape/aquatic	b05cbc5
opentracker	http://erdgeist.org/arts/software/opentracker/	110868e

* Both the mio- and glommio-based implementations were tested.

Hardware

Hetzner CCX62: 48 vCPUs (AMD Milan Epyc 7003)

Software

Software	Version
Debian	Bullseye
Linux	5.10.0-9 sysctl -w net.core.rmem_max=104857600 sysctl -w net.core.rmem_default=104857600
rustc	1.56.1
gcc	10.2.1
aquatic	Run with ./scripts/run-aquatic-udp.sh
opentracker	Before building, run: sed -i "s/^OPTS_production=-O3/OPTS_production=-O3 -march=native -mtune=native/g" Makefile

The default load test configuration was used, except that *duration* was set to 90 seconds and *weight_announce* was set to 5.

Default tracker settings were used, expect that aquatic_udp *recv_buffer* was set to 104857600 and opentracker was configured only to listen to udp.

Joakim Frostegård, 2021-11-08

Measurements

Best results per total worker tier are marked in bold.

aquatic_udp (mio)

aquatic_udp total workers	aquatic_udp socket workers	aquatic_udp request_workers	load test socket workers	load test request workers	responses per second	notes
2	1	1	10	1	**313k**	
3	2	1	10	1	**508k**	
3	2	1	10	1	553k	with tracker cpu pinning
4	3	1	13	3	735k	
4	3	1	10	1	**759k**	
5	4	1	10	1	838k	
5	4	1	13	3	**866k**	
6	4	2	10	1	886k	
6	5	1	13	3	882k	
6	5	1	10	1	**896k**	tracker CPU 475%, load test CPU 1100%
7	5	2	10	1	984k	
7	5	2	13	1	**996k**	
8	6	2	10	1	911k	500%, 1100%
8	6	2	13	3	**1020k**	570%, 1530%
10	7	3	13	3	1019k	
10	8	2	13	3	**1023k**	590%, 1530%
12	10	2	13	3	around 1000k	
12	10	2	10	1	**1017k**	625%, 1100%
14	12	2	13	3	**967k**	630%, 1500%
16	14	2	10	1	**971k**	

Joakim Frostegård, 2021-11-08

aquatic_udp (glommio)

With tracker CPU pinning

aquatic_udp total workers	aquatic_udp socket workers	aquatic_udp request_workers	load test socket workers	load test request workers	responses per second	notes
6	5	1	13	3	around 690k	worse than without CPU pinning
8	6	2	13	3	around 880k	
10	8	2	13	3	1162k	somewhat better than without CPU pinning
12	10	2	13	3	1362k	
14	12	2	13	3	1292k	

CPU pinning doesn't consistently improve throughput. Results were not included in the final comparison.

Without CPU pinning

aquatic_udp total workers	aquatic_udp socket workers	aquatic_udp request_workers	load test socket workers	load test request workers	responses per second	notes
2	1	1	10	1	**200k**	185%, 1100%
3	2	1	10	1	**403k**	300%, 1100%
4	3	1	10	1	419k	300% (!), 1100%
4	3	1	13	3	**668k**	
5	4	1	10	1	672k	385%, 1100%
5	4	1	13	3	**716k**	450 %
6	4	2	10	1	around 540k	
6	5	1	10	1	**834k**	440 %
7	6	1	10	1	780k	
7	6	1	13	3	742k	
7	5	2	13	3	**945k**	
8	6	2	10	1	763k	600%, 1100%
8	7	1	10	1	around 670k	

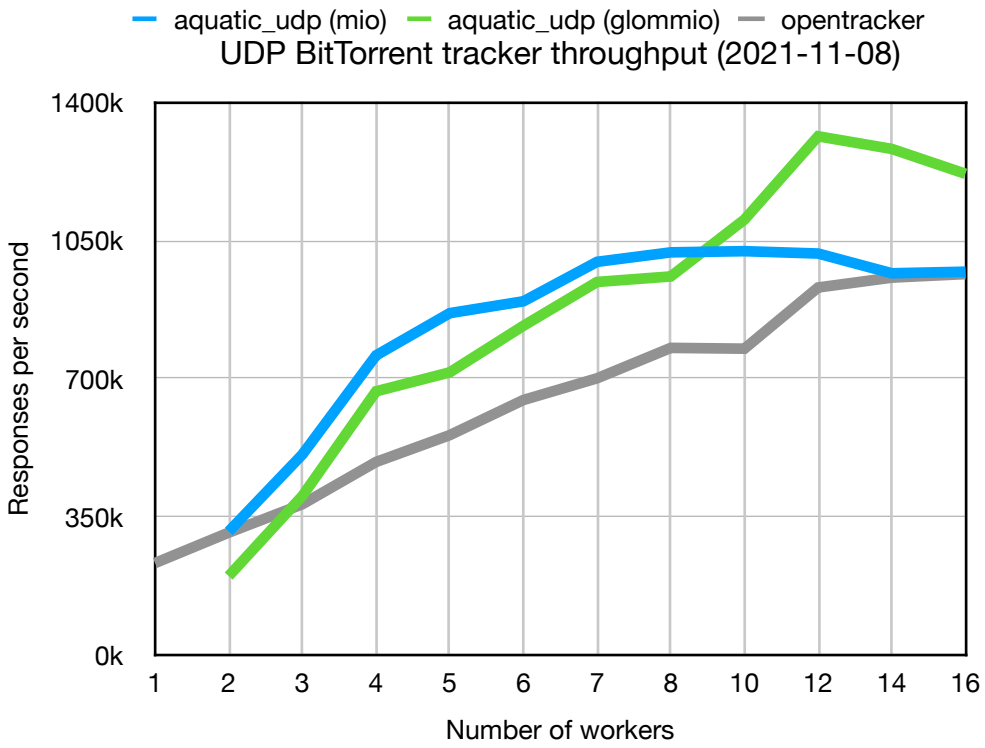
aquatic_udp total workers	aquatic_udp socket workers	aquatic_udp request_workers	load test socket workers	load test request workers	responses per second	notes
8	6	2	13	3	**959k**	700 %
9	7	2	10	1	962k	700%, 1100%
9	7	2	13	3	**1165k**	
10	8	2	10	1	912k	700%, 1100%
10	8	2	13	3	**1103k**	
12	10	2	13	3	1154k	
12	10	2	10	1	1301k	900%, 1100%
12	10	2	10	1	**1314k**	
14	12	2	10	1	1160k	
14	12	2	13	3	1237k	
14	12	2	13	3	**1282k**	850%, 1400%
14	11	3	13	3	1256k	1080%, 1400%
16	14	2	13	3	**1219k**	

opentracker

workers	load test socket workers	load test request workers	responses per second	notes
1	10	1	206k	event mode
1	10	1	**233k**	100%, 1100%
2	10	1	**310k**	
3	10	1	**382k**	380%, 1100%
4	13	3	486k	
4	10	1	**489k**	
5	10	1	**557k**	475%. 1090%
6	13	3	635k	565%, 1520%
6	10	1	**646k**	565%, 1090%
7	10	1	**701k**	
8	13	3	661k	

workers	load test socket workers	load test request workers	responses per second	notes
8	10	1	**778k**	
10	13	3	772k	895%, 1520%
10	10	1	**776k**	815%, 1090%
12	13	3	866k	
12	10	1	**931k**	
14	10	1	**956k**	
16	10	1	**965k**	

Results



- Both aquatic implementations significantly outperform opentracker in throughput when using 4 or more workers. When using 2 workers, opentracker performs the same as aquatic (mio) and better than aquatic (glommio).
- aquatic (mio) performs better than aquatic (glommio) when using up to 8 workers, after which the latter takes the lead
- aquatic (mio) throughput peaks at just over 1 million responses per second, with 8 workers
- aquatic (glommio) throughput peaks at 1.3 million responses per second, with 12 workers
- opentracker throughput peaks at just shy of 1 million responses per second, with 16 workers

Joakim Frostegård, 2021-11-08

UDP BitTorrent tracker throughput (2021-11-08)

Number of workers	aquatic_udp (mio)	aquatic_udp (glommio)	opentracker
1	n/a	n/a	**233**
2	**313**	200	310
3	**508**	403	382
4	**759**	668	489
5	**866**	716	557
6	**896**	834	646
7	**996**	945	701
8	**1020**	959	778
10	1023	**1103**	776
12	1017	**1314**	931
14	967	**1282**	956
16	971	**1219**	965

Results are shown in thousands of responses per second. The best result for each number of workers is marked in bold.