

PolePosition

the open source database benchmark

<http://www.polepos.org>

Participating teams

db4o

the open source object database for Java and .NET

<http://www.db4o.com>

db4o

the open source object database for Java and .NET

<http://www.db4o.com>

db4o SNAPSHOT

the open source object database for Java and .NET

<http://www.db4o.com>

db4o LAZY

the open source object database for Java and .NET

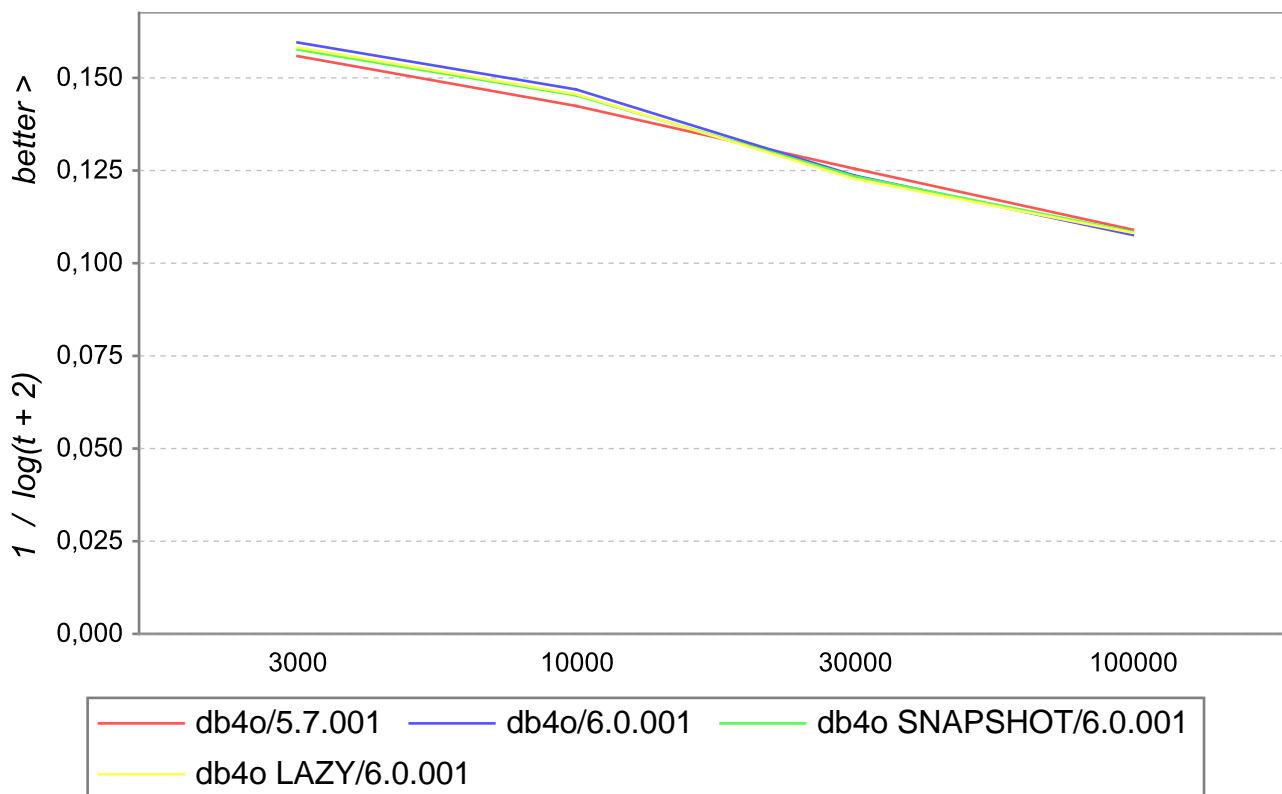
<http://www.db4o.com>

Circuit: Melbourne

writes, reads and deletes unstructured flat objects of one kind in bulk mode

Lap: write

t [time in ms]	objects:3000	objects:10000	objects:30000	objects:100000
db4o/5.7.001	609	1118	2886	9617
db4o/6.0.001	525	903	3251	10852
db4o	567	973	3302	10104
db4o	554	959	3449	10317

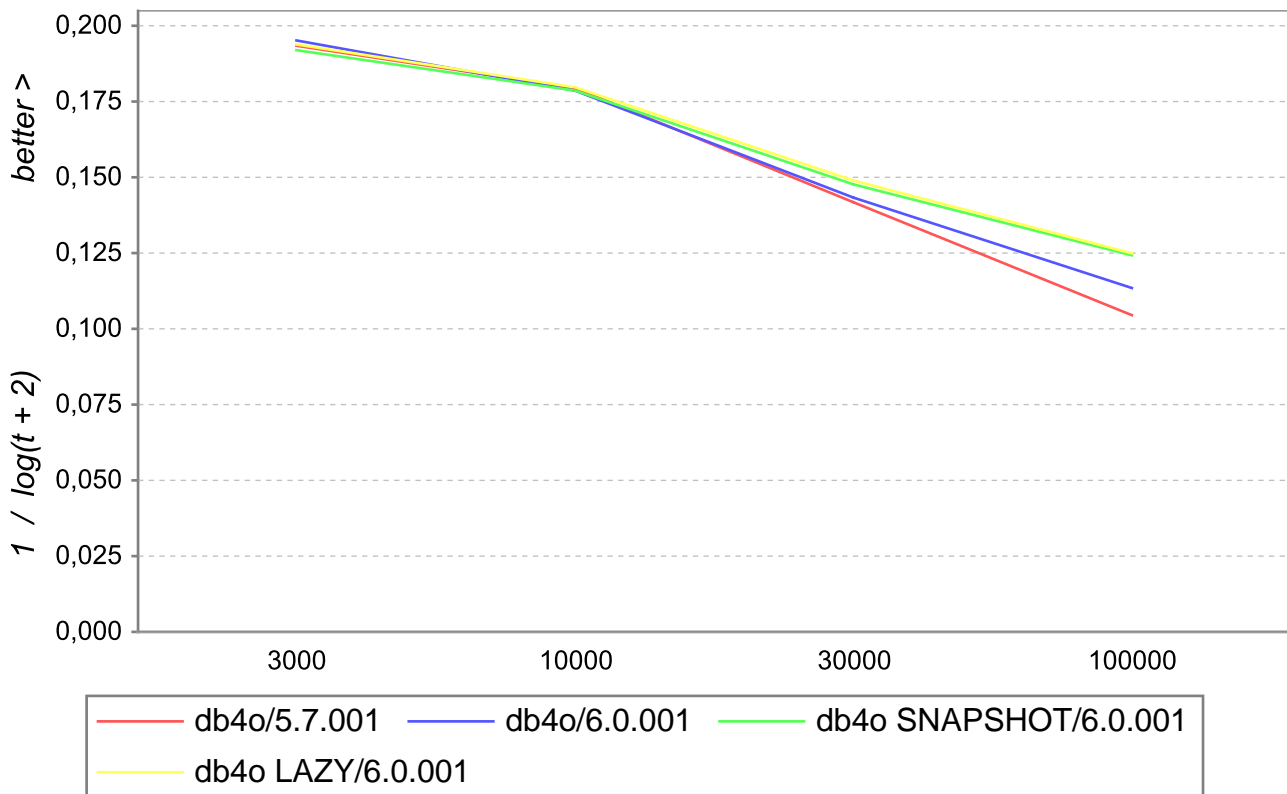


Circuit: Melbourne

writes, reads and deletes unstructured flat objects of one kind in bulk mode

Lap: read

t [time in ms]	objects:3000	objects:10000	objects:30000	objects:100000
db4o/5.7.001	174	261	1155	14294
db4o/6.0.001	166	268	1071	6716
db4o	181	268	871	3122
db4o	172	259	822	2991

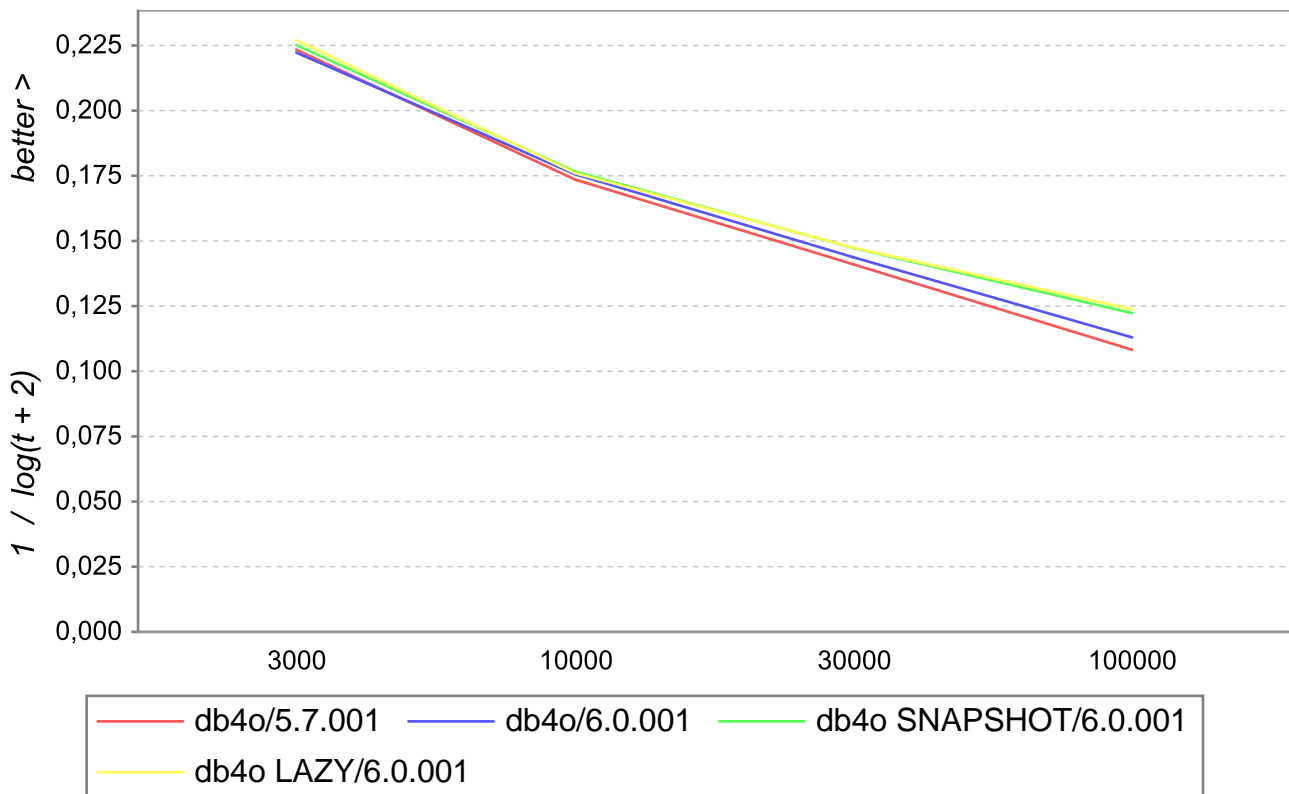


Circuit: Melbourne

writes, reads and deletes unstructured flat objects of one kind in bulk mode

Lap: read_hot

t [time in ms]	objects:3000	objects:10000	objects:30000	objects:100000
db4o/5.7.001	86	316	1200	10272
db4o/6.0.001	88	295	1049	6962
db4o	83	286	888	3556
db4o	80	291	883	3225

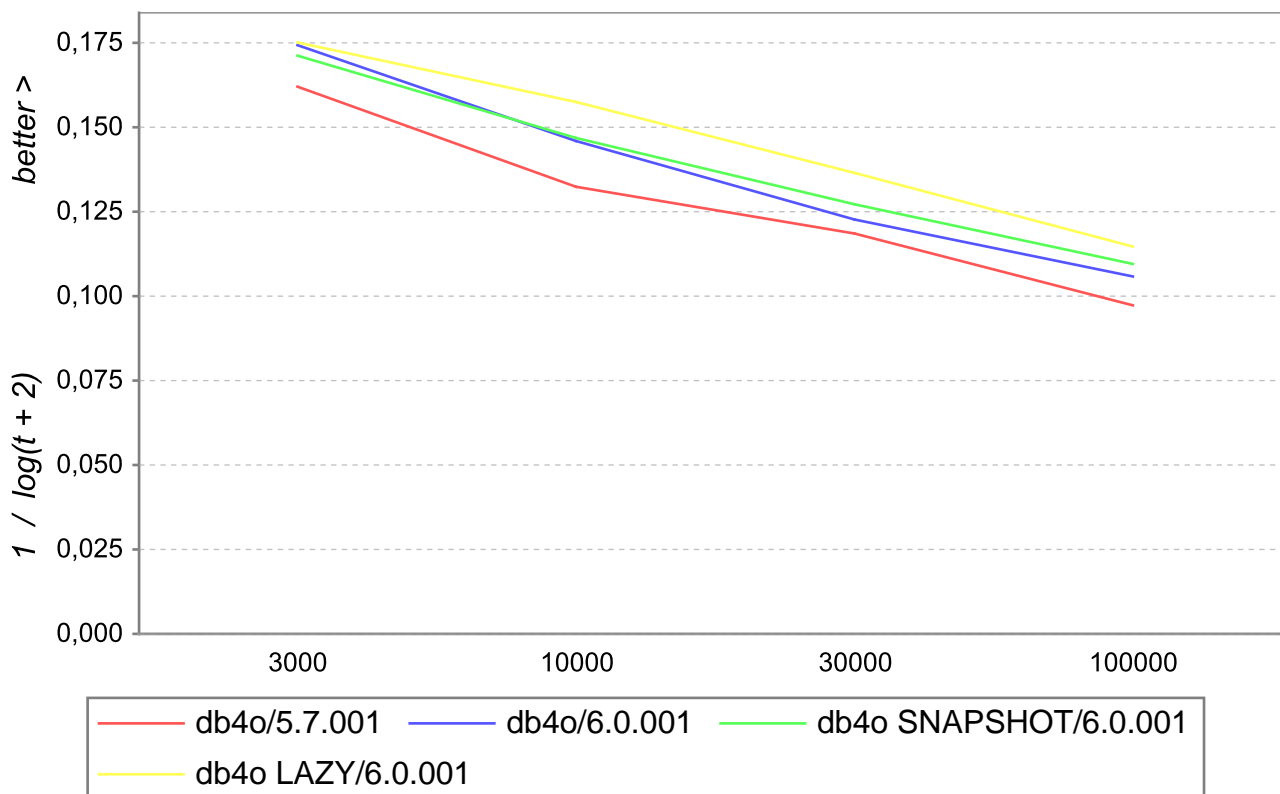


Circuit: Melbourne

writes, reads and deletes unstructured flat objects of one kind in bulk mode

Lap: delete

t [time in ms]	objects:3000	objects:10000	objects:30000	objects:100000
db4o/5.7.001	477	1906	4605	29122
db4o/6.0.001	308	944	3462	12701
db4o	342	905	2593	9217
db4o	300	570	1517	6121

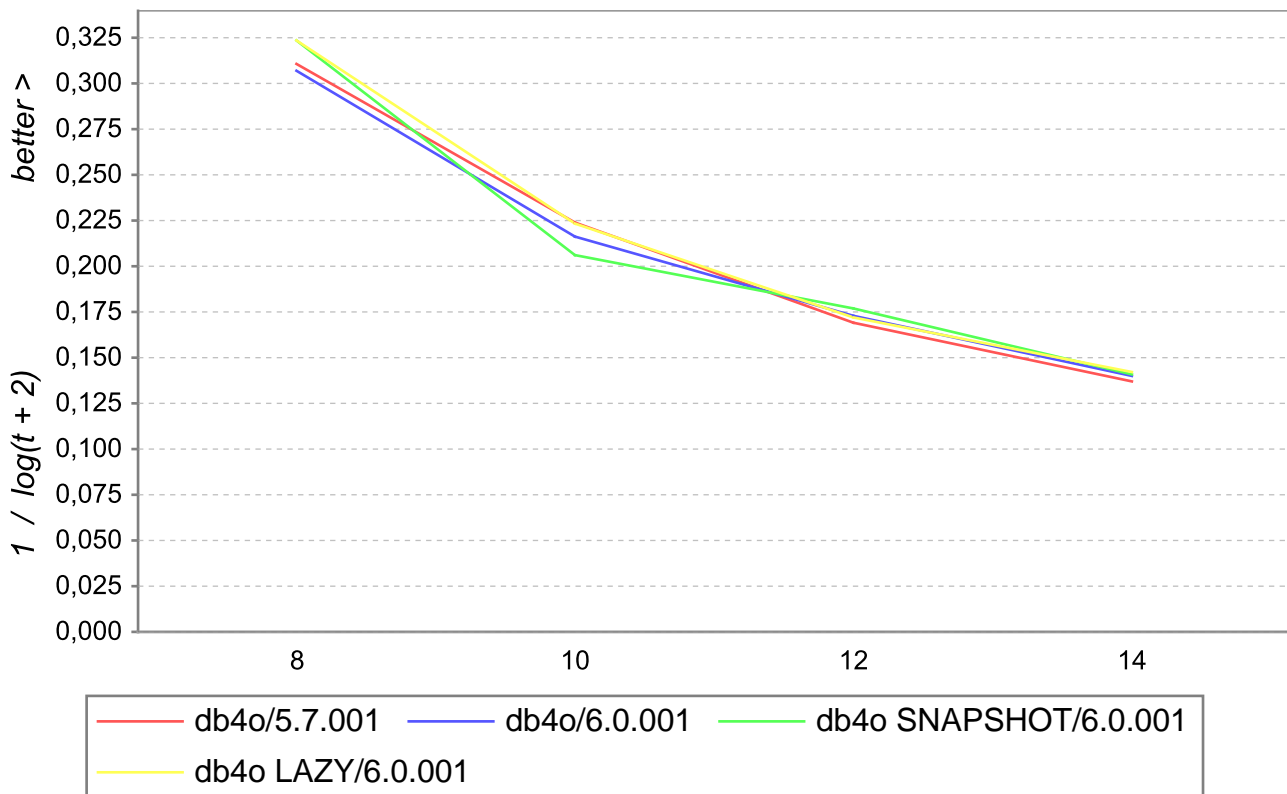


Circuit: Sepang

writes, reads and then deletes an object tree

Lap: write

t [time in ms]	depth:8	depth:10	depth:12	depth:14
db4o/5.7.001	23	85	368	1476
db4o/6.0.001	24	100	323	1262
db4o	20	126	283	1209
db4o	20	86	333	1134

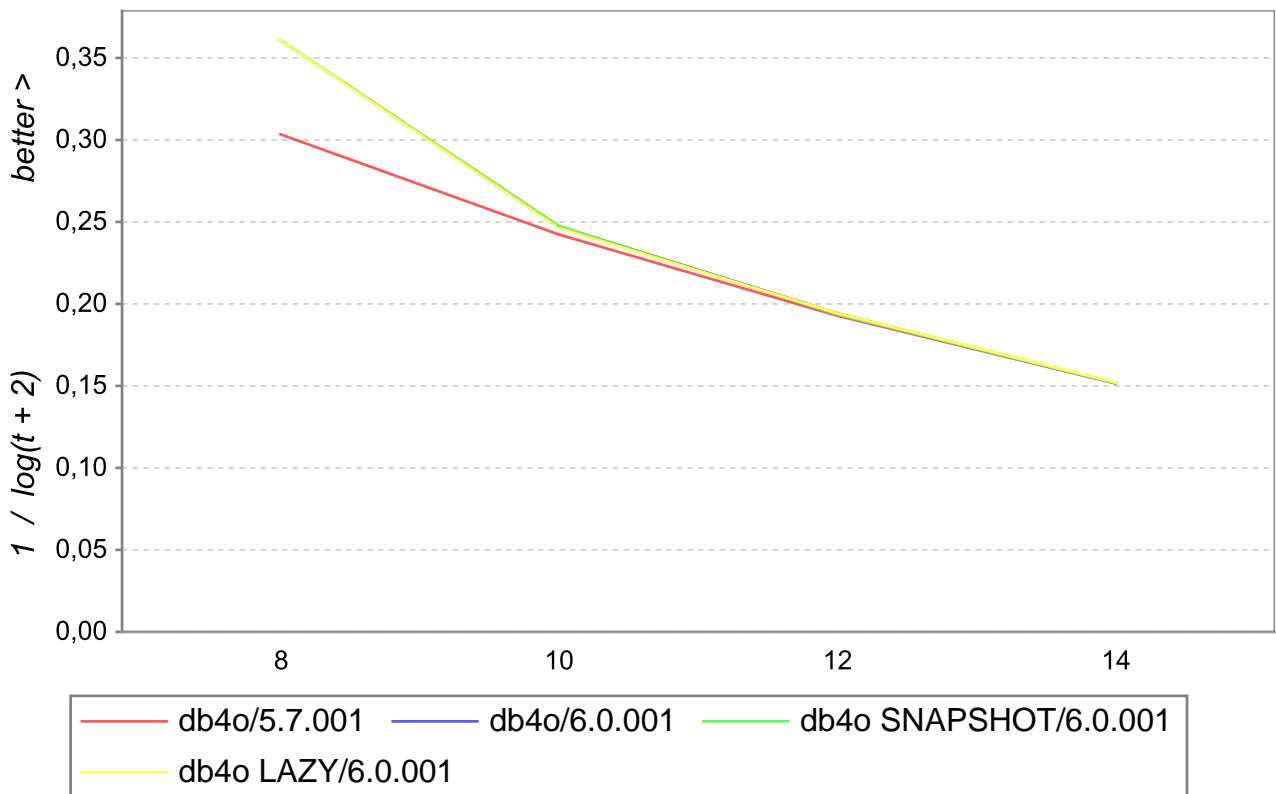


Circuit: Sepang

writes, reads and then deletes an object tree

Lap: read

t [time in ms]	depth:8	depth:10	depth:12	depth:14
db4o/5.7.001	25	60	177	737
db4o/6.0.001	14	55	170	727
db4o	14	55	173	730
db4o	14	56	170	717

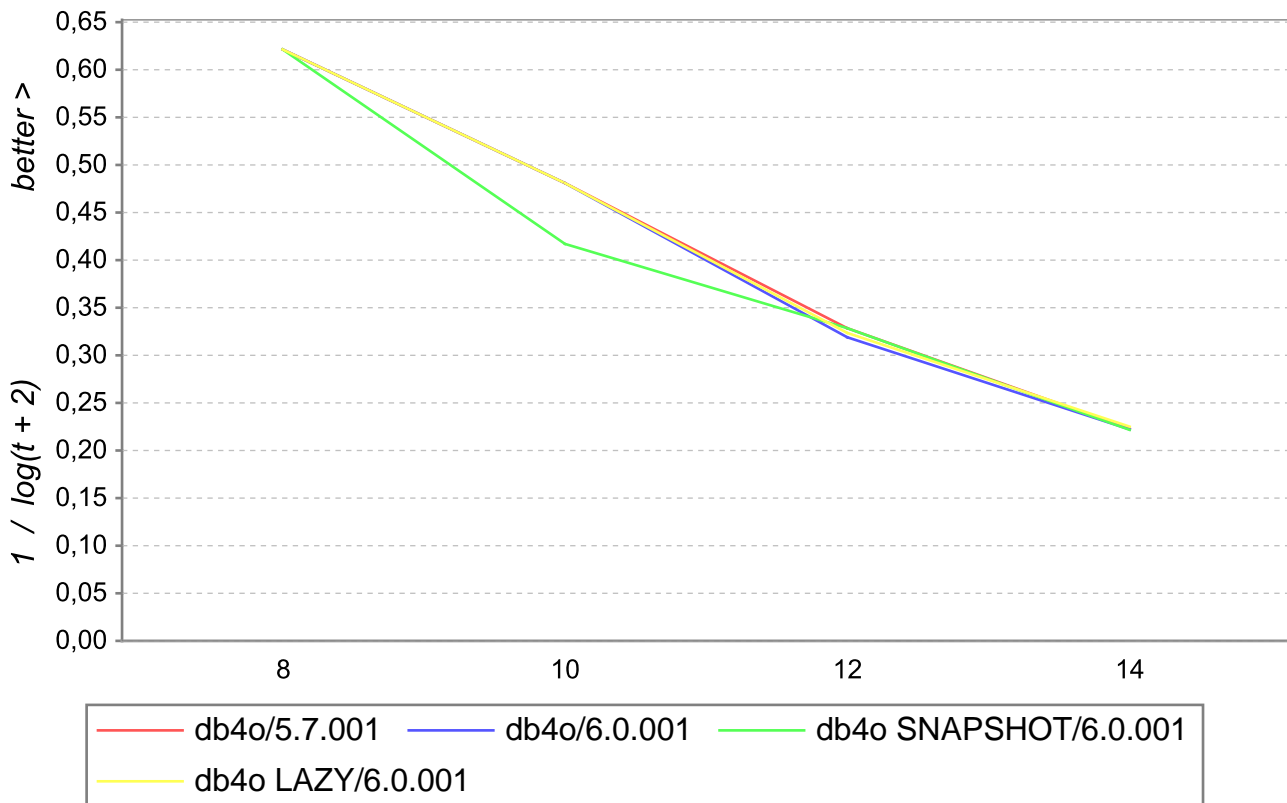


Circuit: Sepang

writes, reads and then deletes an object tree

Lap: read_hot

t [time in ms]	depth:8	depth:10	depth:12	depth:14
db4o/5.7.001	3	6	19	87
db4o/6.0.001	3	6	21	88
db4o	3	9	19	89
db4o	3	6	20	83

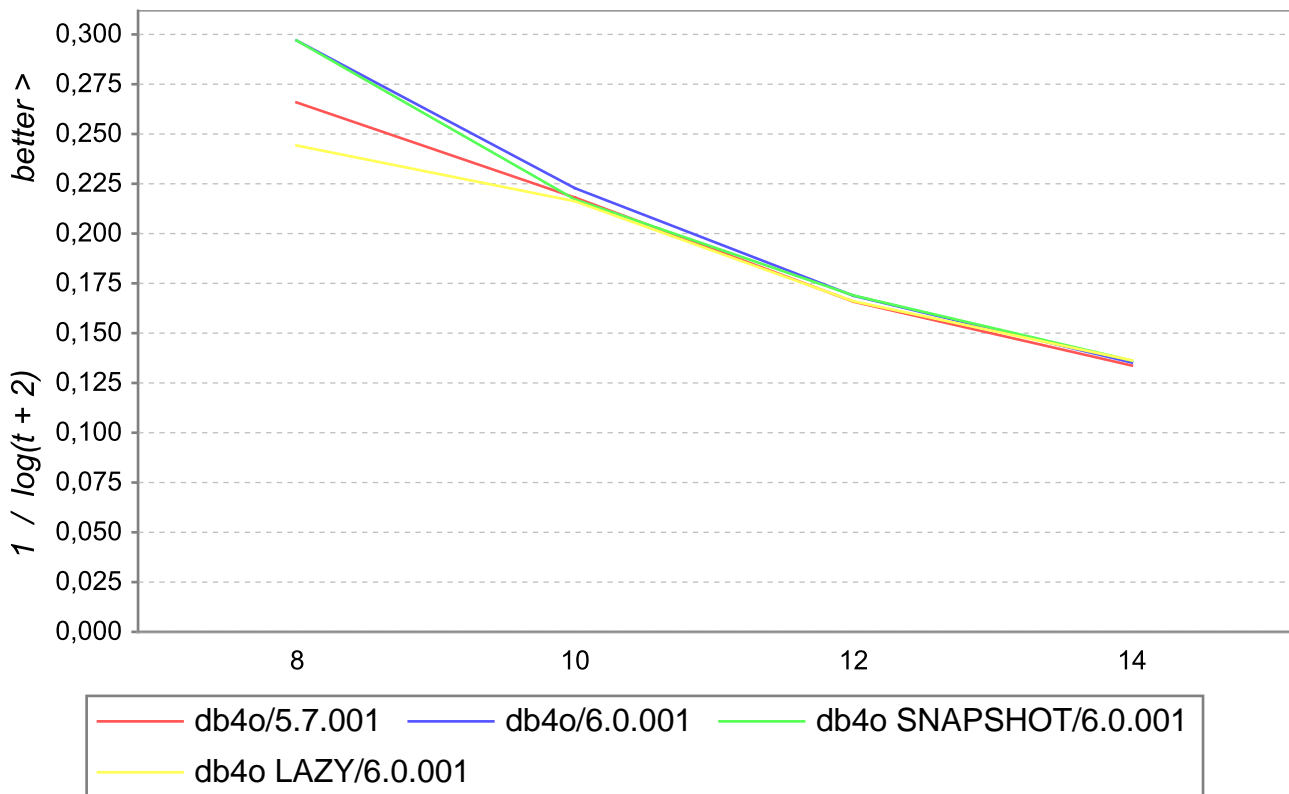


Circuit: Sepang

writes, reads and then deletes an object tree

Lap: delete

t [time in ms]	depth:8	depth:10	depth:12	depth:14
db4o/5.7.001	41	96	414	1765
db4o/6.0.001	27	87	372	1624
db4o	27	98	370	1540
db4o	58	100	411	1529

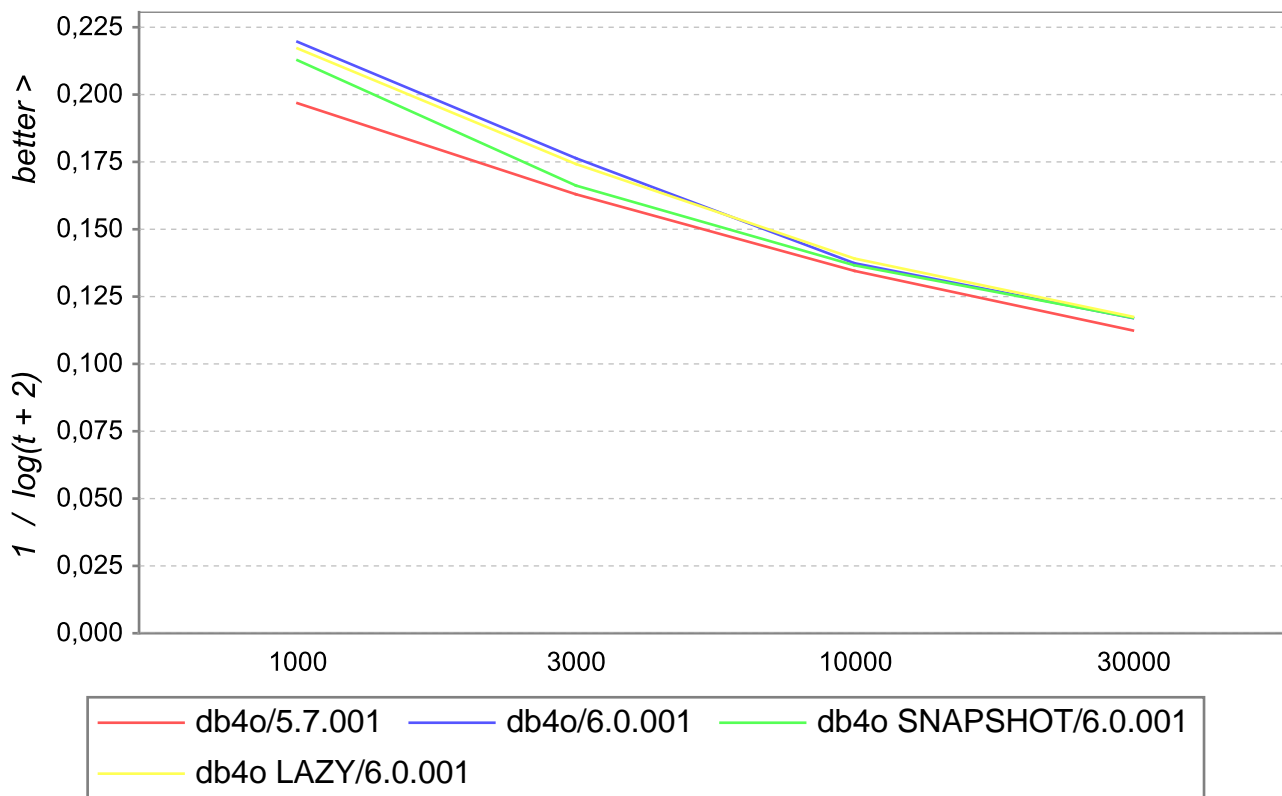


Circuit: Bahrain

write, query, update and delete simple flat objects individually

Lap: write

t [time in ms]	selects:900 objects:1000 updates:100	selects:900 objects:3000 updates:100	selects:900 objects:10000 updates:100	selects:900 objects:30000 updates:100
db4o/5.7.001	159	460	1688	7301
db4o/6.0.001	93	288	1447	5145
db4o	108	408	1508	5101
db4o	98	309	1321	4962

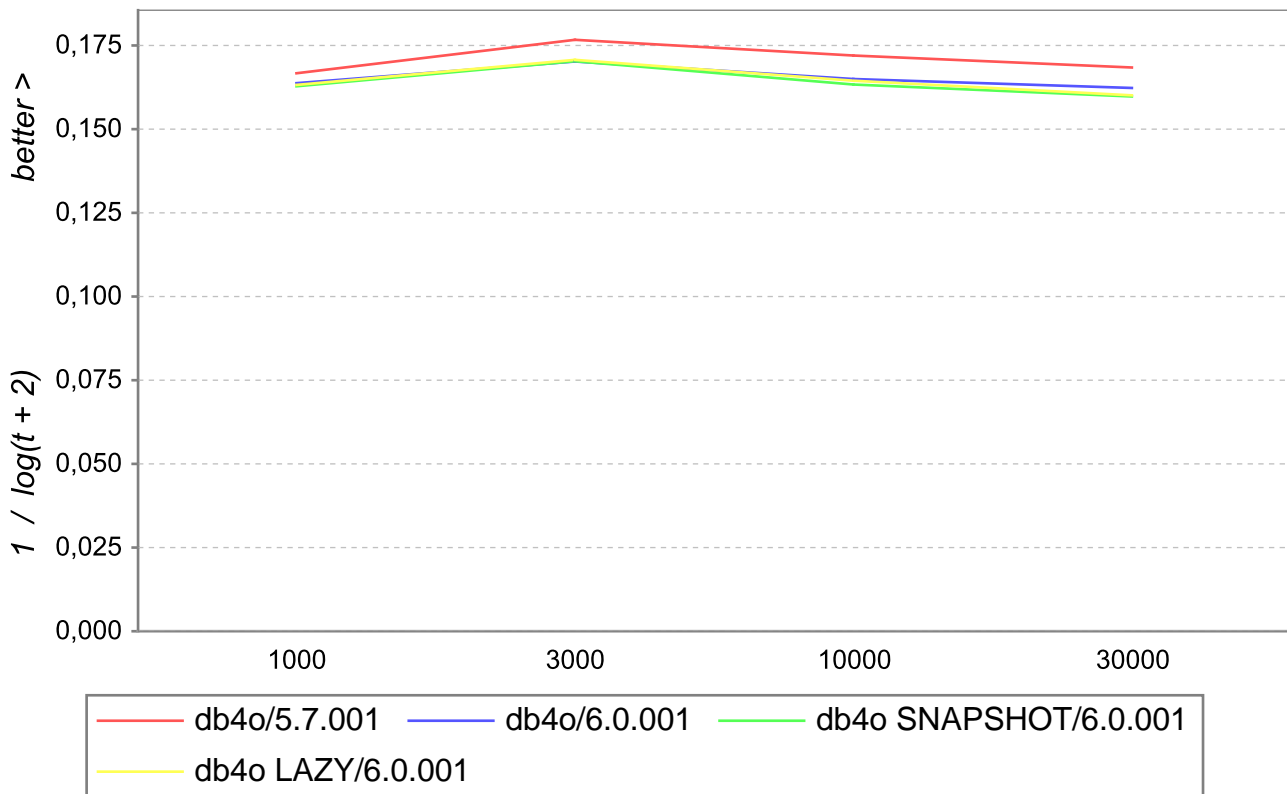


Circuit: Bahrain

write, query, update and delete simple flat objects individually

Lap: query_indexed_string

t [time in ms]	selects:900 objects:1000 updates:100	selects:900 objects:3000 updates:100	selects:900 objects:10000 updates:100	selects:900 objects:30000 updates:100
db4o/5.7.001	401	285	333	377
db4o/6.0.001	447	354	427	472
db4o	463	353	454	521
db4o	456	348	435	514

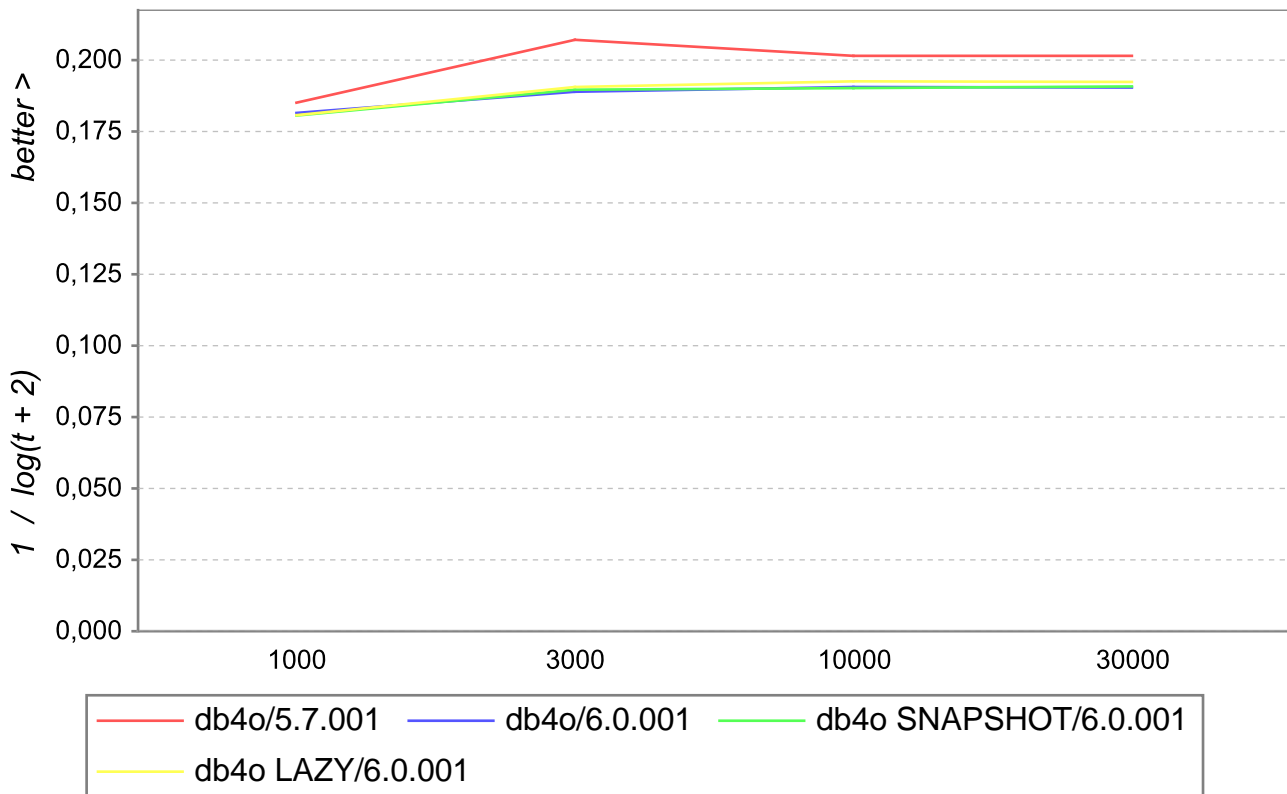


Circuit: Bahrain

write, query, update and delete simple flat objects individually

Lap: query_indexed_int

t [time in ms]	selects:900 objects:1000 updates:100	selects:900 objects:3000 updates:100	selects:900 objects:10000 updates:100	selects:900 objects:30000 updates:100
db4o/5.7.001	220	123	141	141
db4o/6.0.001	245	197	188	189
db4o	252	193	190	187
db4o	251	188	178	179

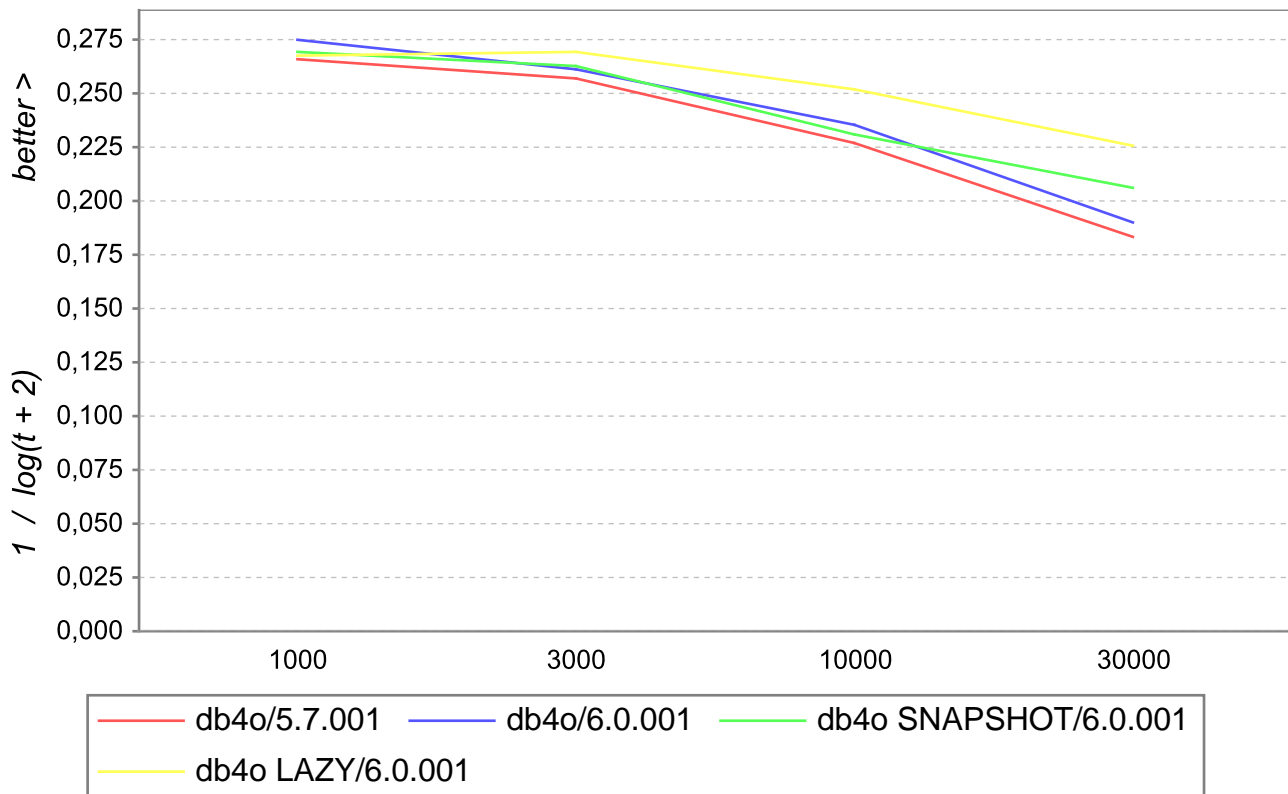


Circuit: Bahrain

write, query, update and delete simple flat objects individually

Lap: update

t [time in ms]	selects:900 objects:1000 updates:100	selects:900 objects:3000 updates:100	selects:900 objects:10000 updates:100	selects:900 objects:30000 updates:100
db4o/5.7.001	41	47	80	232
db4o/6.0.001	36	44	68	191
db4o	39	43	74	126
db4o	40	39	51	82

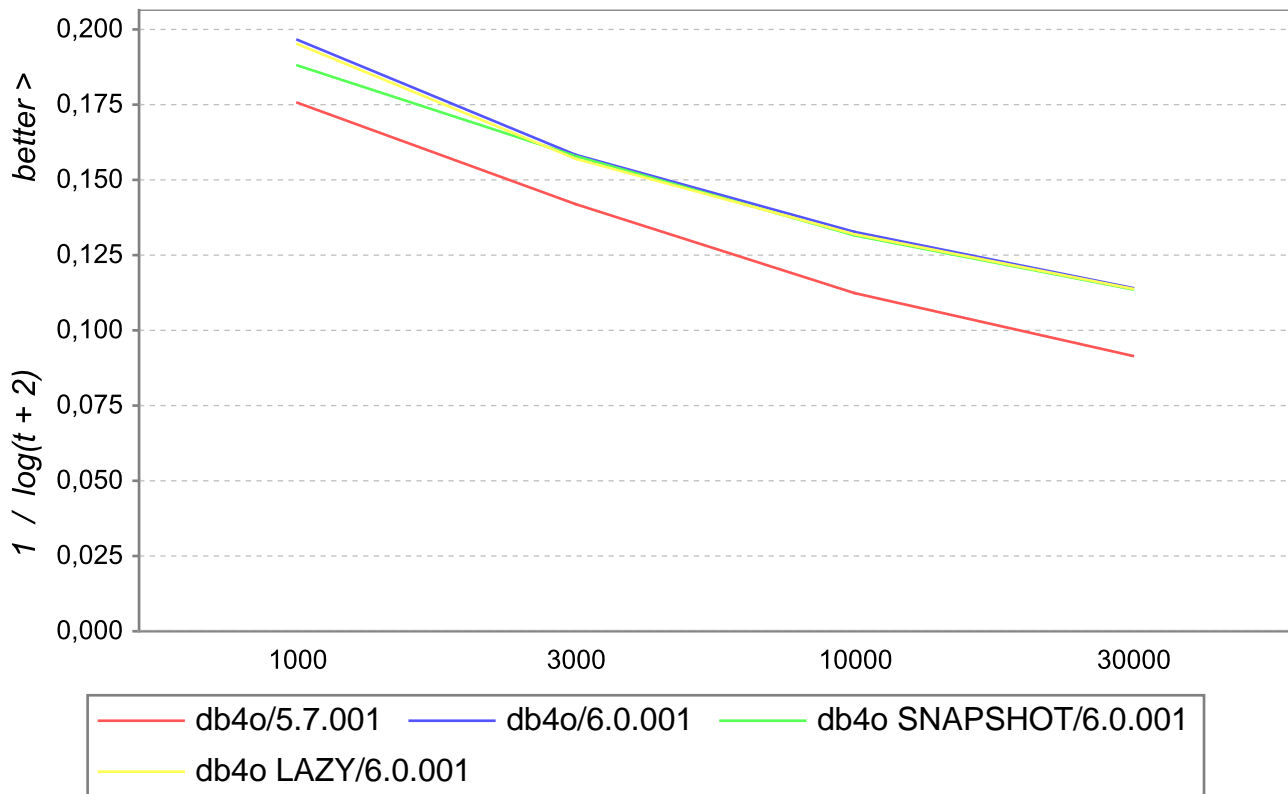


Circuit: Bahrain

write, query, update and delete simple flat objects individually

Lap: delete

t [time in ms]	selects:900 objects:1000 updates:100	selects:900 objects:3000 updates:100	selects:900 objects:10000 updates:100	selects:900 objects:30000 updates:100
db4o/5.7.001	295	1147	7312	55790
db4o/6.0.001	160	552	1869	6417
db4o	202	561	1998	6654
db4o	166	582	1955	6549

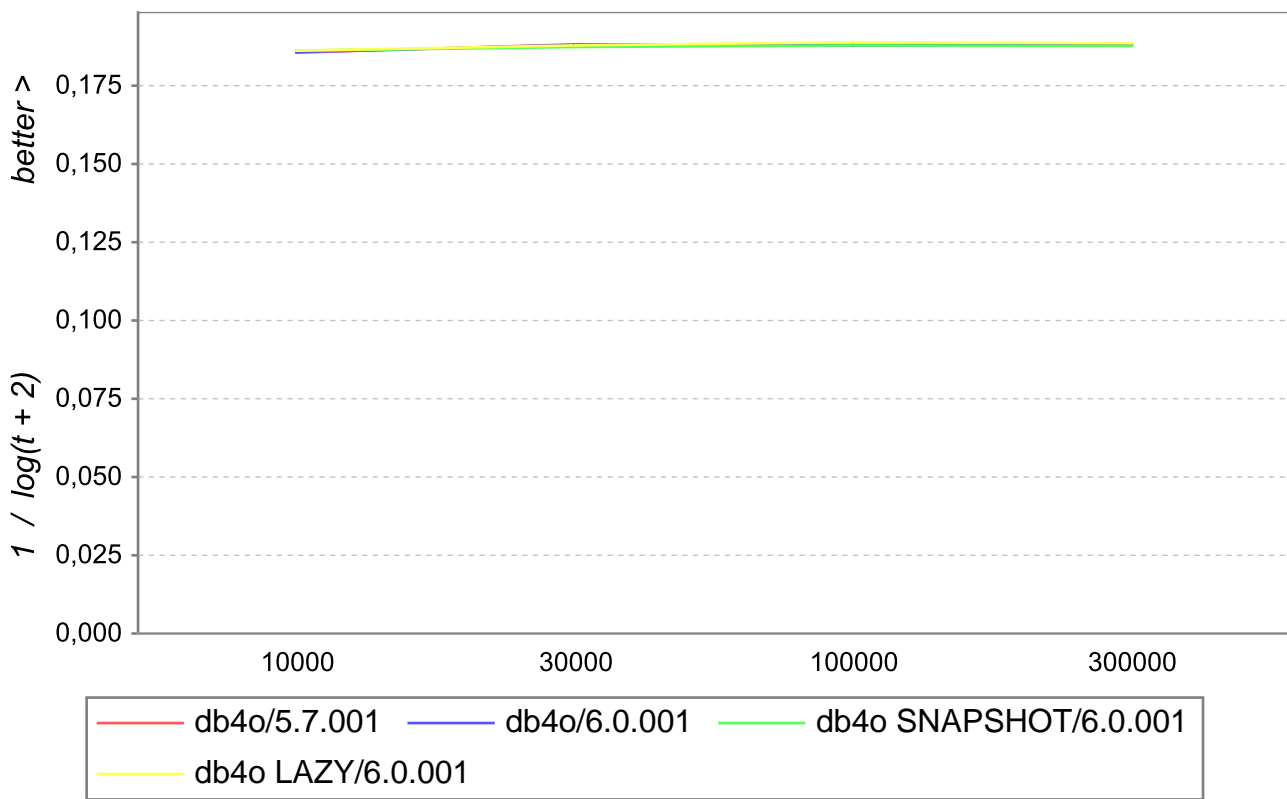


Circuit: Imola

retrieves objects by native id

Lap: retrieve

t [time in ms]	selects:5000 objects:10000	selects:5000 objects:30000	selects:5000 objects:100000	selects:5000 objects:300000
db4o/5.7.001	217	201	202	202
db4o/6.0.001	216	202	203	202
db4o	213	206	203	204
db4o	213	203	197	199

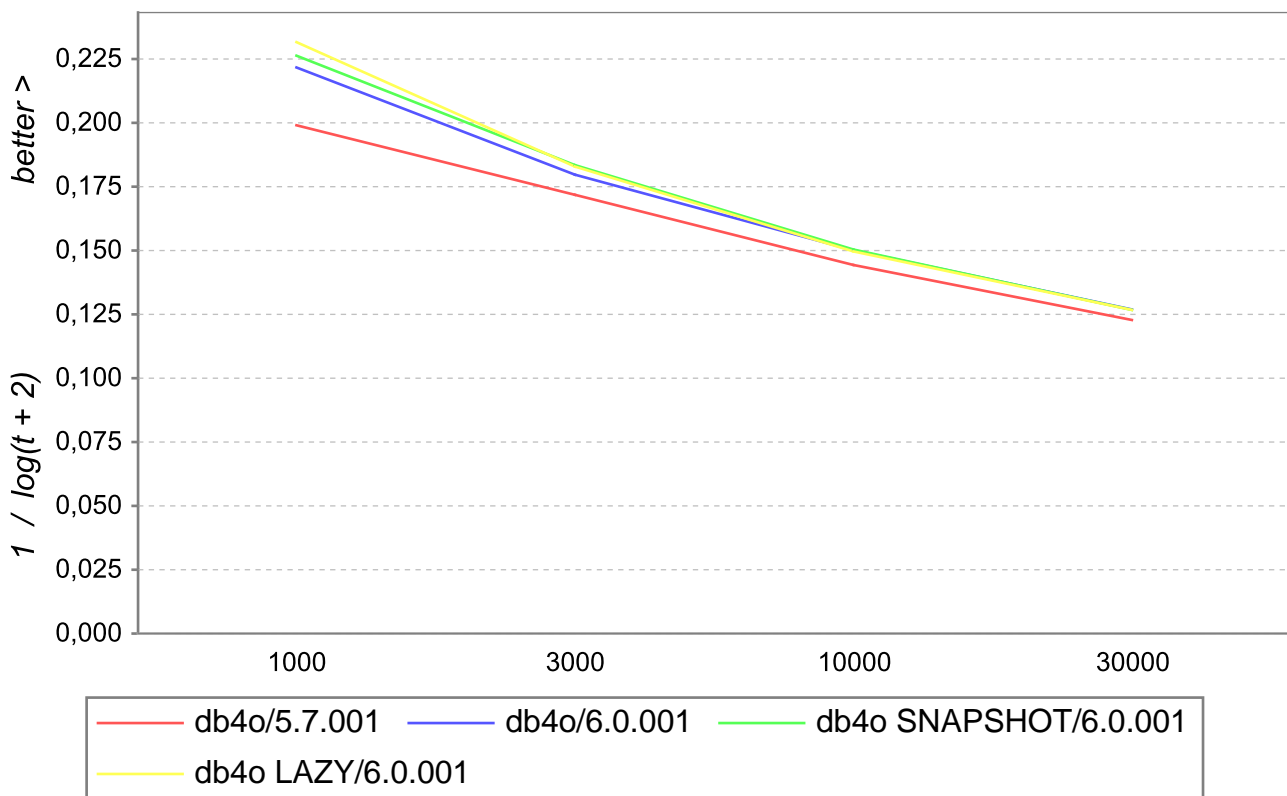


Circuit: Barcelona

writes, reads, queries and deletes objects with a 5 level inheritance structure

Lap: write

t [time in ms]	selects:100 objects:1000	selects:100 objects:3000	selects:100 objects:10000	selects:100 objects:30000
db4o/5.7.001	150	335	1018	3438
db4o/6.0.001	89	259	780	2638
db4o	81	231	770	2666
db4o	73	235	796	2668

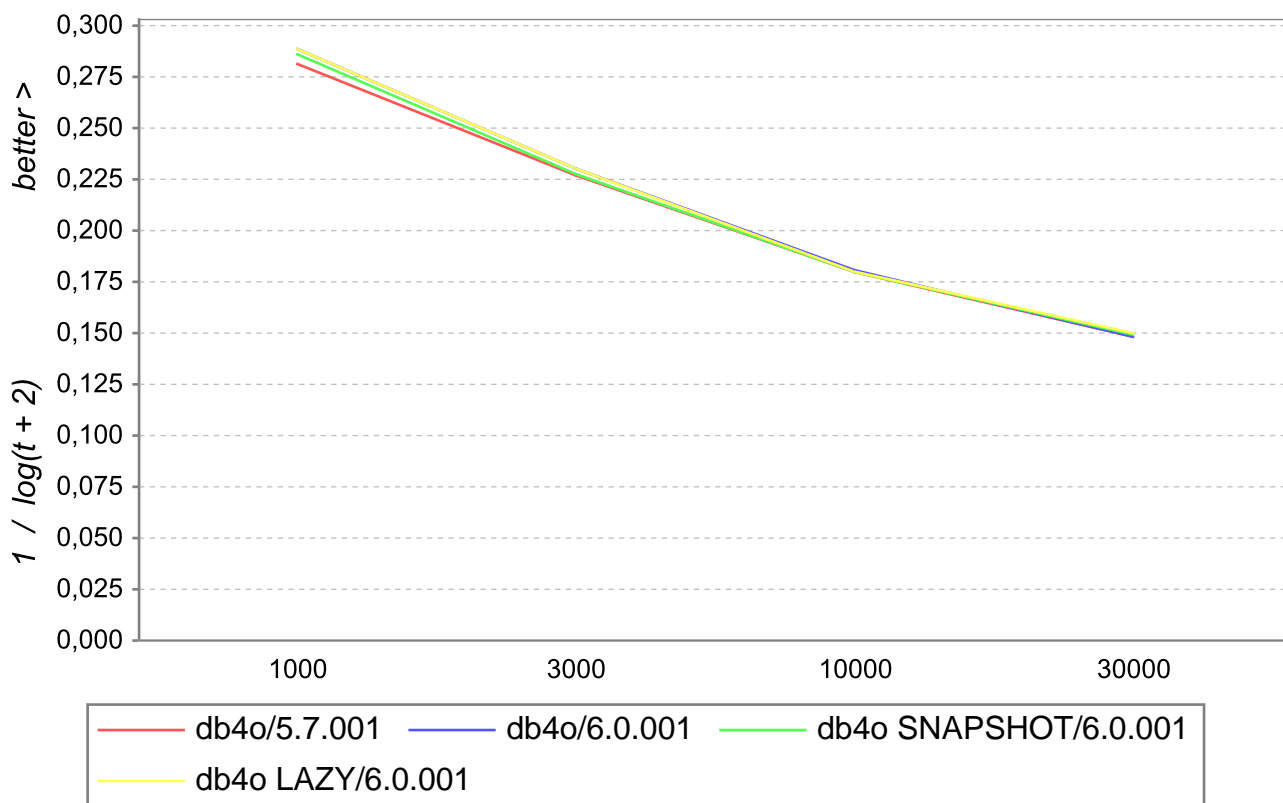


Circuit: Barcelona

writes, reads, queries and deletes objects with a 5 level inheritance structure

Lap: read

t [time in ms]	selects:100 objects:1000	selects:100 objects:3000	selects:100 objects:10000	selects:100 objects:30000
db4o/5.7.001	33	80	258	848
db4o/6.0.001	30	75	251	849
db4o	31	79	258	811
db4o	30	75	257	783

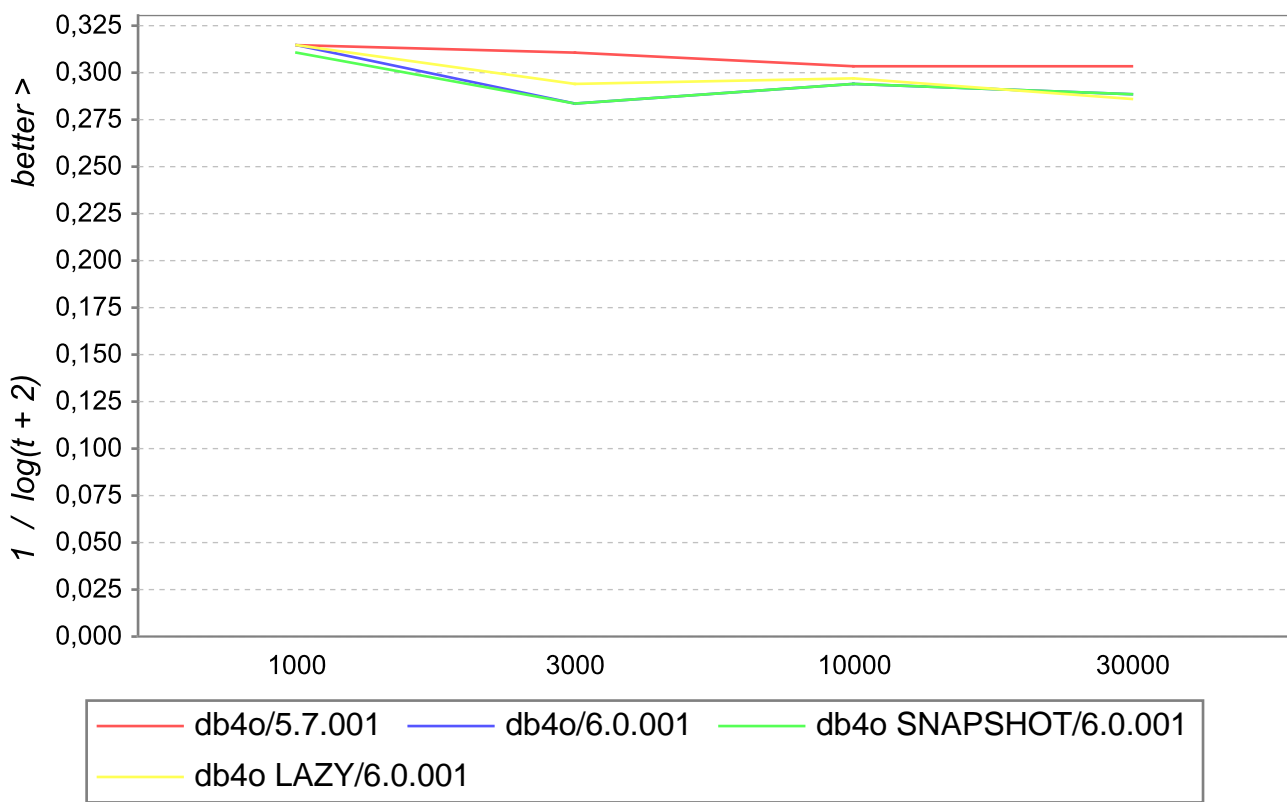


Circuit: Barcelona

writes, reads, queries and deletes objects with a 5 level inheritance structure

Lap: query

t [time in ms]	selects:100 objects:1000	selects:100 objects:3000	selects:100 objects:10000	selects:100 objects:30000
db4o/5.7.001	22	23	25	25
db4o/6.0.001	22	32	28	30
db4o	23	32	28	30
db4o	22	28	27	31

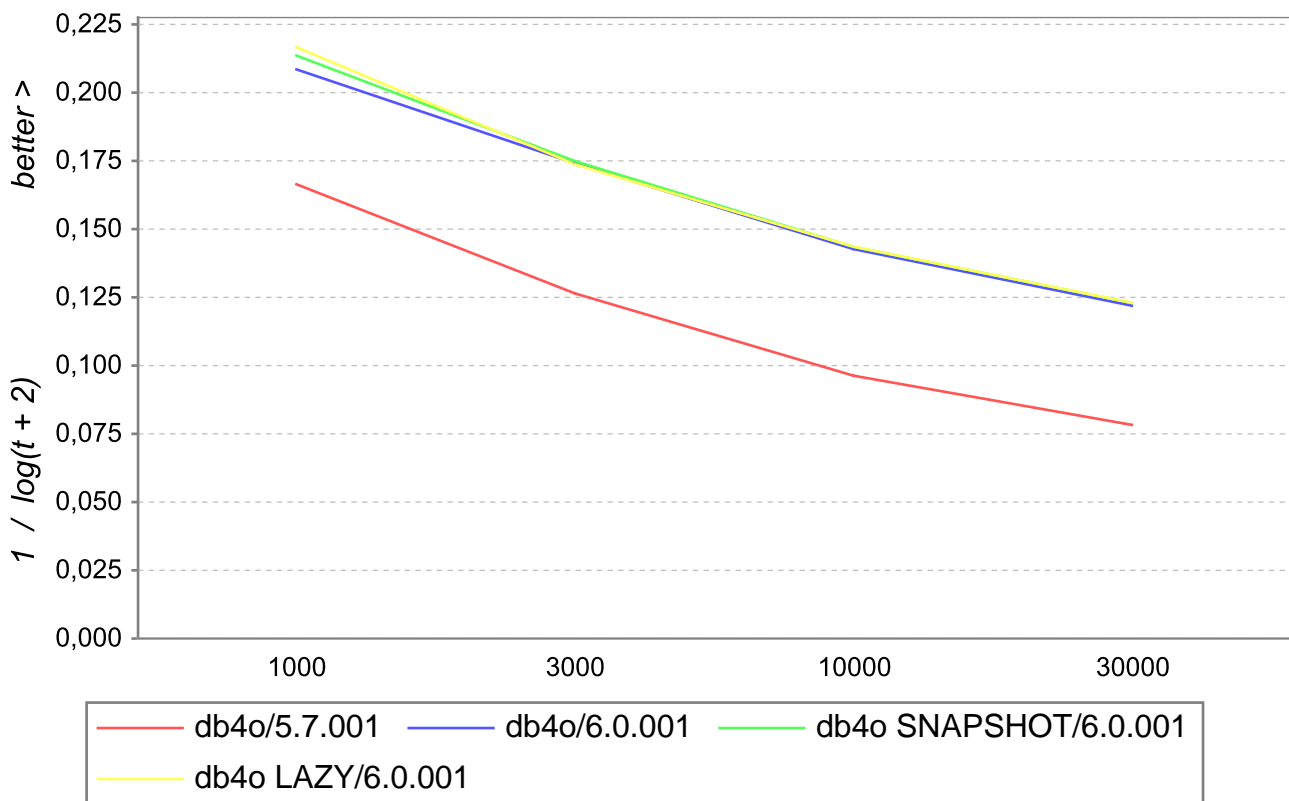


Circuit: Barcelona

writes, reads, queries and deletes objects with a 5 level inheritance structure

Lap: delete

t [time in ms]	selects:100 objects:1000	selects:100 objects:3000	selects:100 objects:10000	selects:100 objects:30000
db4o/5.7.001	405	2718	32272	352768
db4o/6.0.001	119	310	1103	3639
db4o	106	302	1060	3403
db4o	99	314	1056	3386

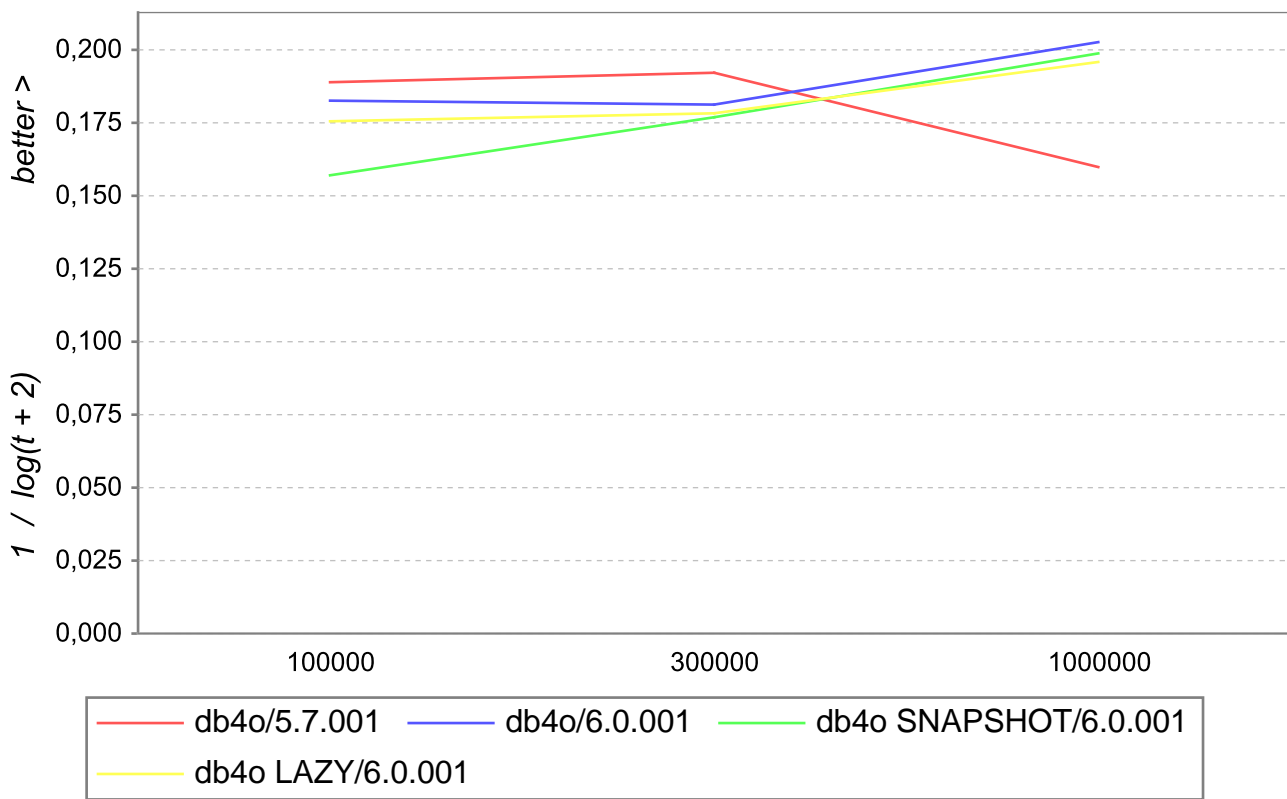


Circuit: Monaco

tests the scalability of commit performance

Lap: commits

t [time in ms]	commits:30 objects:100000	commits:30 objects:300000	commits:30 objects:1000000
db4o/5.7.001	197	180	519
db4o/6.0.001	237	247	137
db4o	581	283	151
db4o LAZY/6.0.001	296	271	163

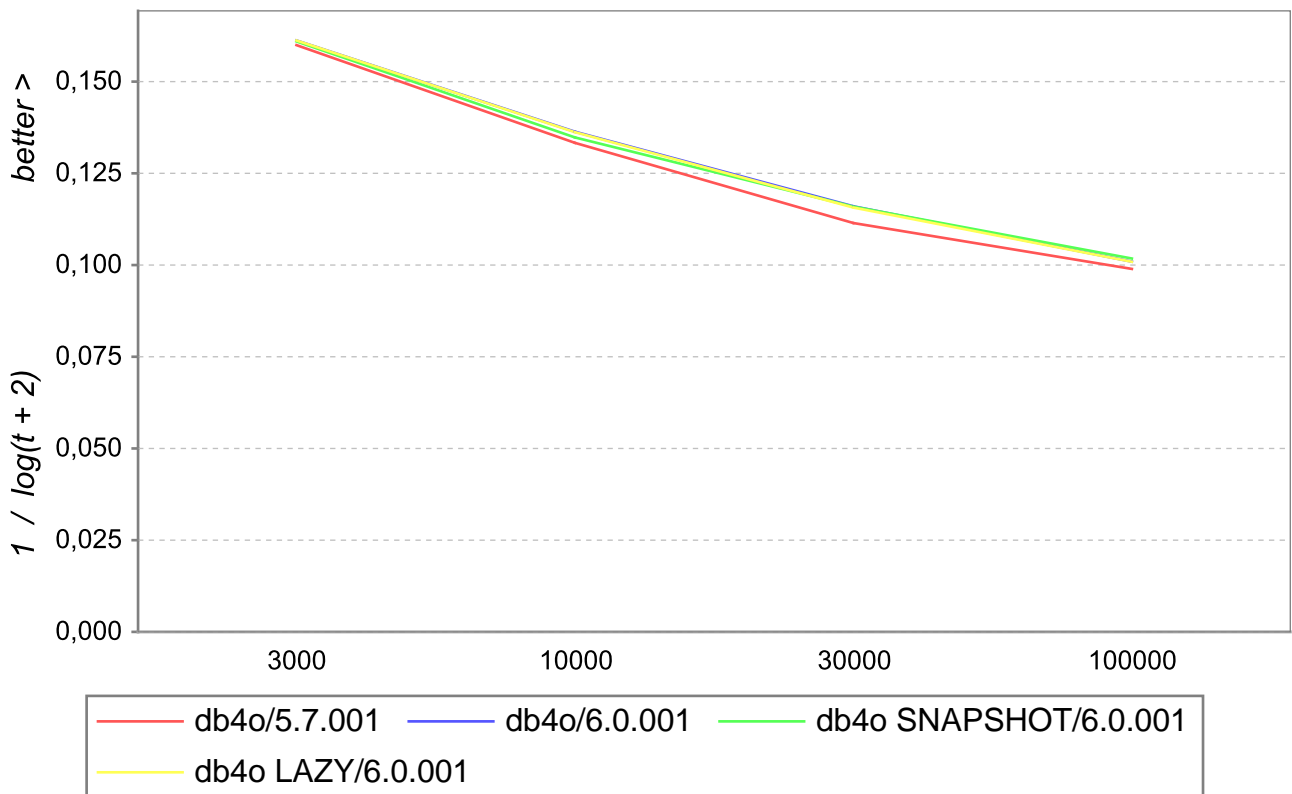


Circuit: Nurburgring

tests String storing and loading efficiency

Lap: write

t [time in ms]	objects:3000	objects:10000	objects:30000	objects:100000
db4o/5.7.001	517	1809	7884	24511
db4o/6.0.001	492	1530	5537	20091
db4o	497	1666	5574	18489
db4o	492	1541	5682	20155

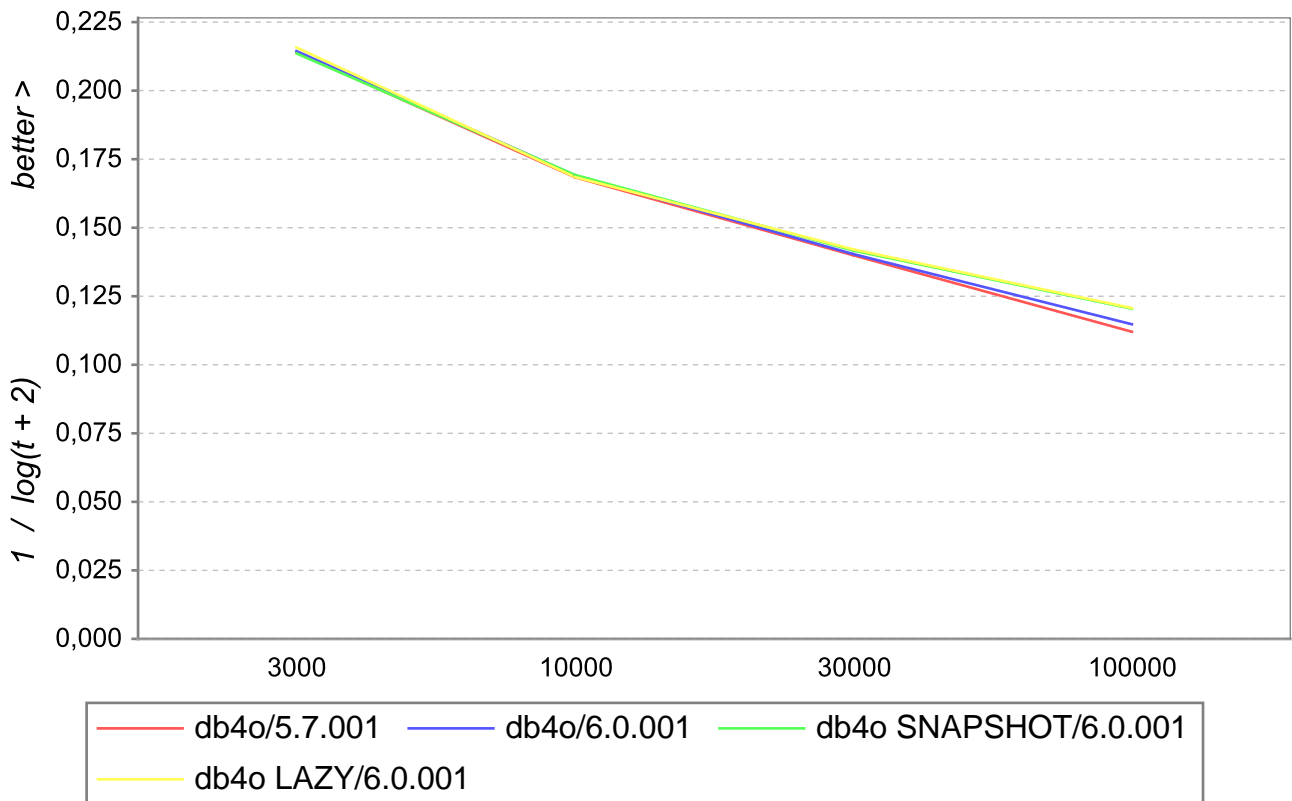


Circuit: Nurburgring

tests String storing and loading efficiency

Lap: read

t [time in ms]	objects:3000	objects:10000	objects:30000	objects:100000
db4o/5.7.001	105	377	1267	7515
db4o/6.0.001	104	368	1239	6074
db4o	106	366	1161	4030
db4o	101	376	1137	3965

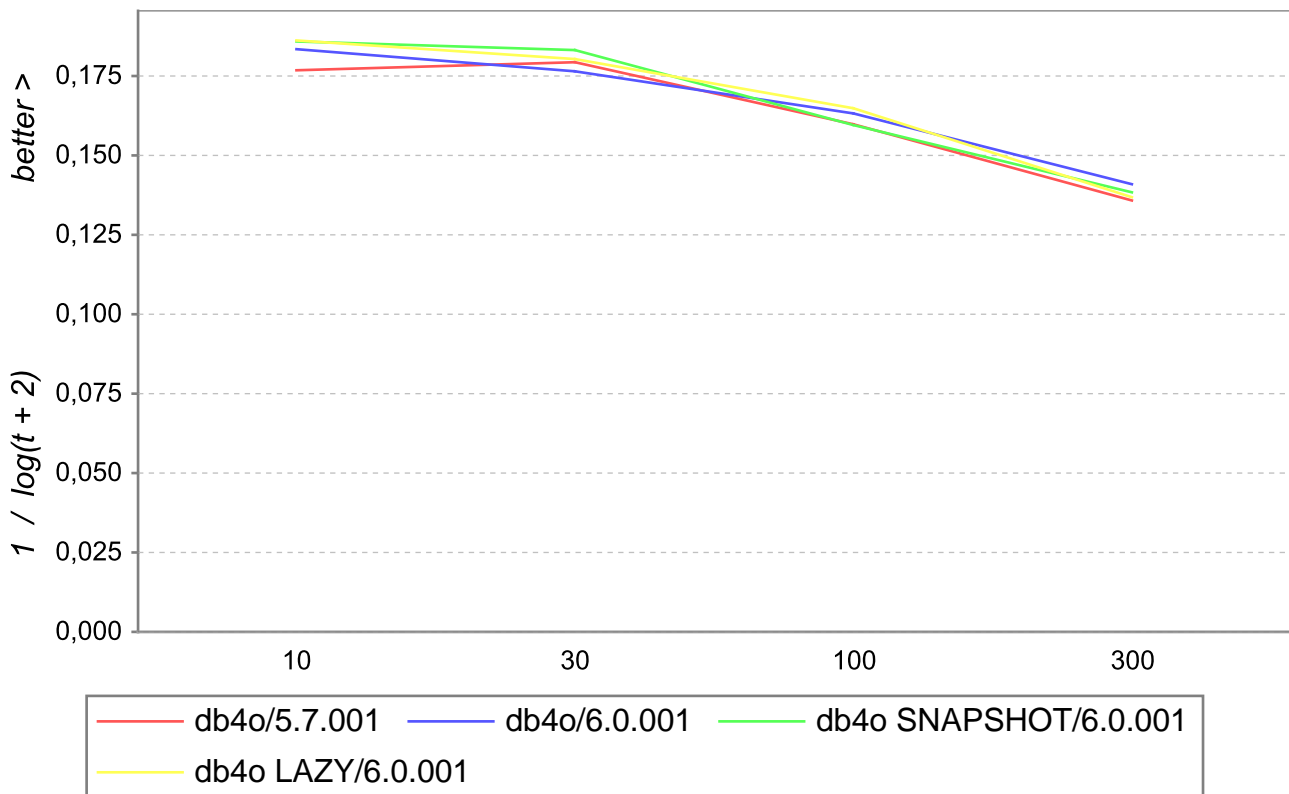


Circuit: Montreal

writes and reads 1000 ArrayLists

Lap: write

t [time in ms]	size:10	size:30	size:100	size:300
db4o/5.7.001	284	262	520	1574
db4o/6.0.001	231	287	456	1204
db4o	215	233	525	1373
db4o	213	254	429	1489

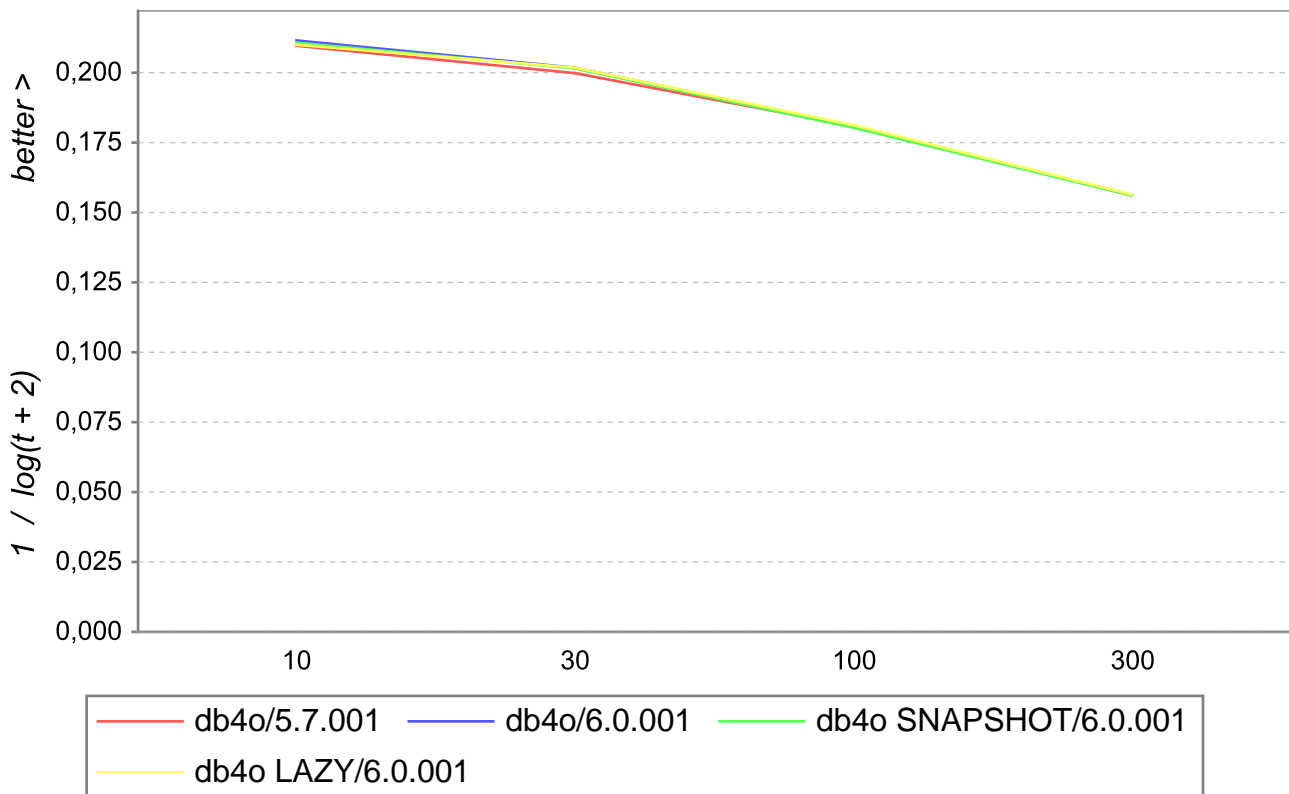


Circuit: Montreal

writes and reads 1000 ArrayLists

Lap: read

t [time in ms]	size:10	size:30	size:100	size:300
db4o/5.7.001	116	147	250	603
db4o/6.0.001	111	140	250	602
db4o	113	141	254	605
db4o	115	140	247	596

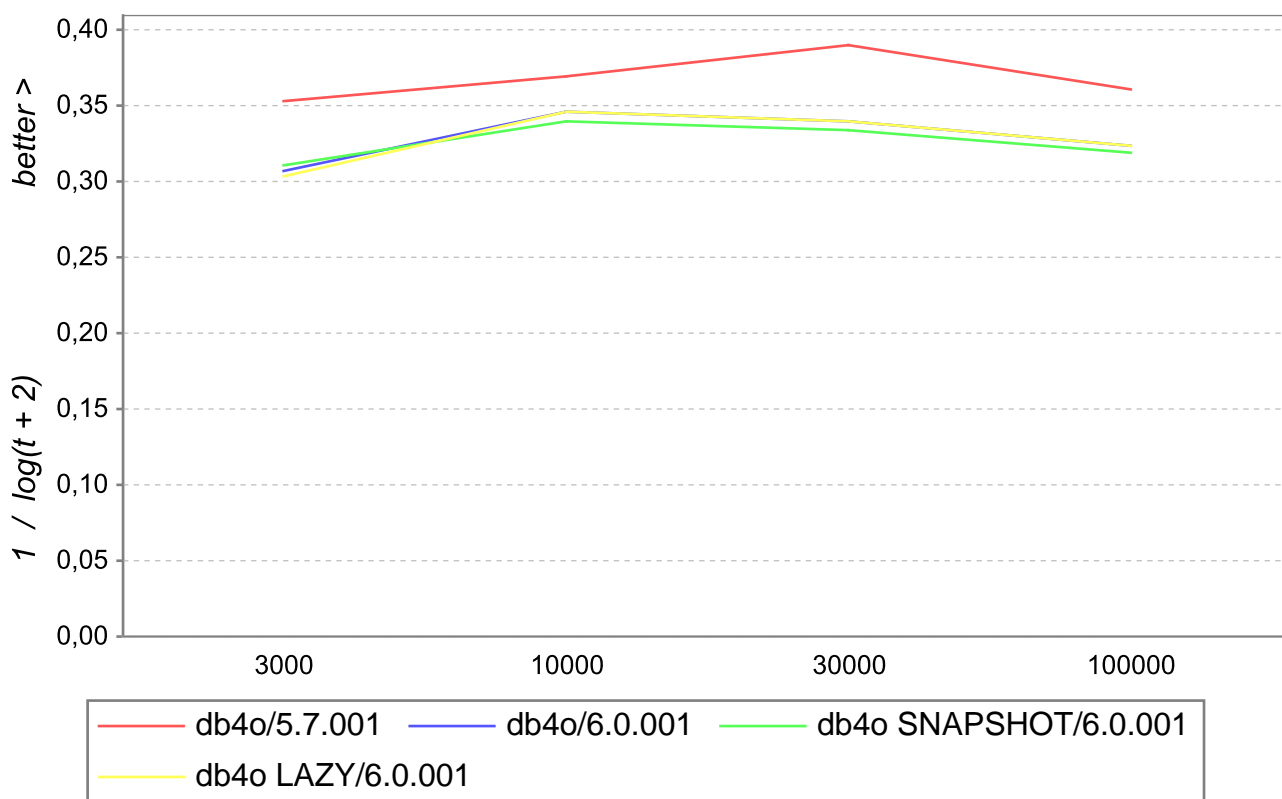


Circuit: Indianapolis

executes a variety of queries to test the efficiency of the query processor

Lap: queryRange

t [time in ms]	selects:30 objects:3000	selects:30 objects:10000	selects:30 objects:30000	selects:30 objects:100000
db4o/5.7.001	15	13	11	14
db4o/6.0.001	24	16	17	20
db4o	23	17	18	21
db4o	25	16	17	20

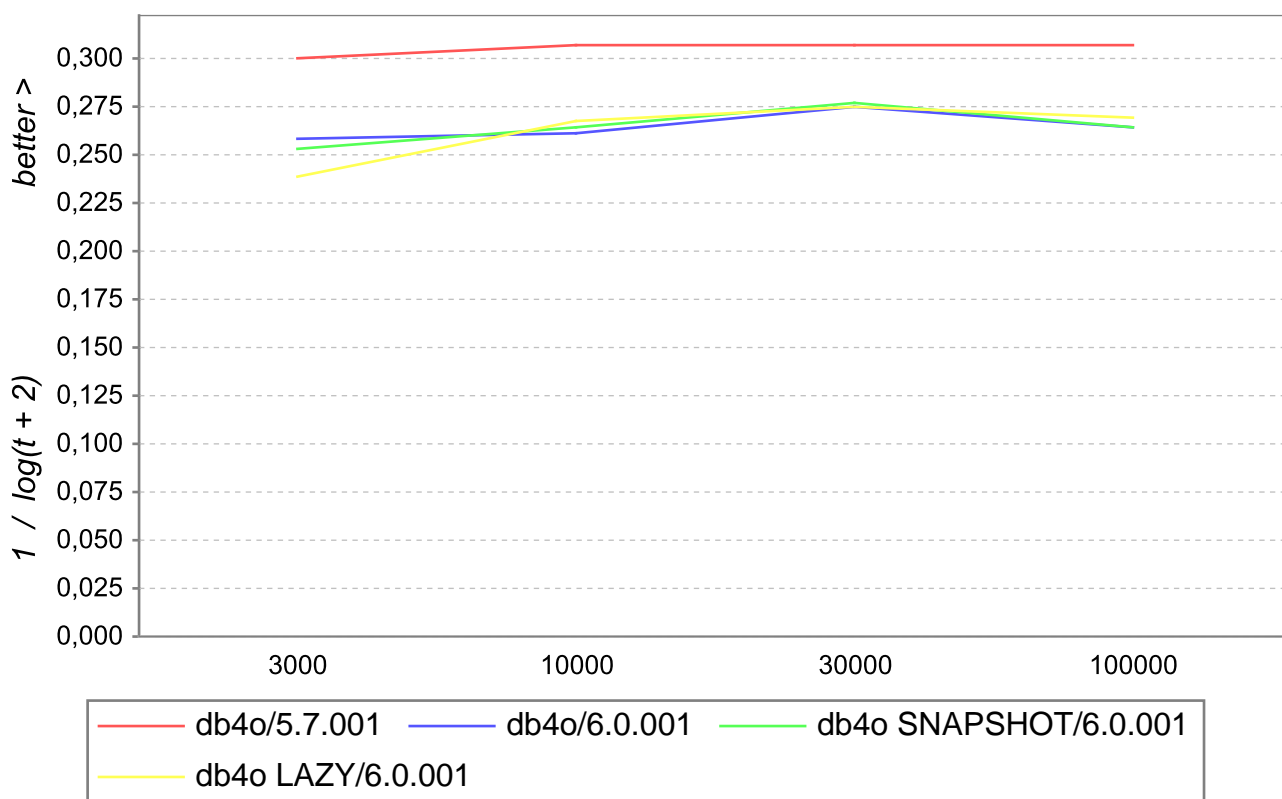


Circuit: Indianapolis

executes a variety of queries to test the efficiency of the query processor

Lap: query5Links

t [time in ms]	selects:30 objects:3000	selects:30 objects:10000	selects:30 objects:30000	selects:30 objects:100000
db4o/5.7.001	26	24	24	24
db4o/6.0.001	46	44	36	42
db4o	50	42	35	42
db4o	64	40	36	39

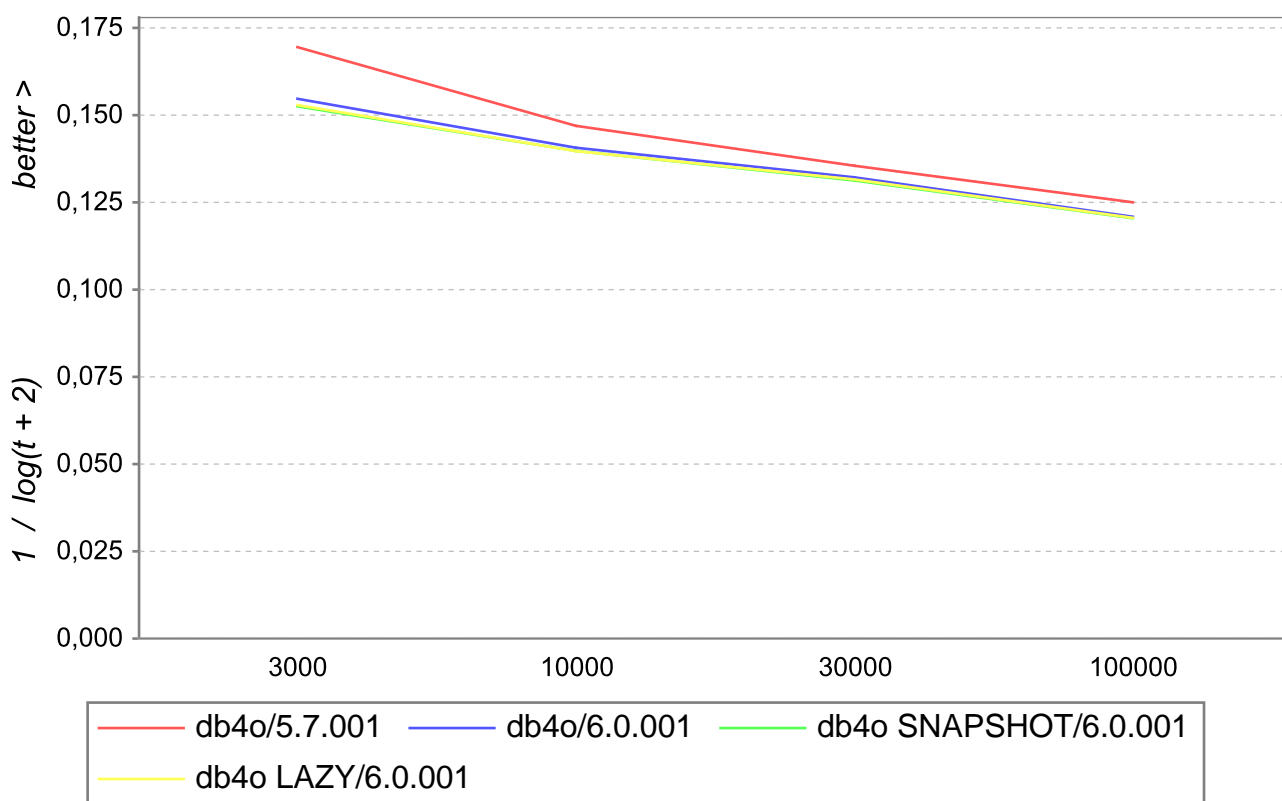


Circuit: Indianapolis

executes a variety of queries to test the efficiency of the query processor

Lap: queryPreferShortPath

t [time in ms]	selects:30 objects:3000	selects:30 objects:10000	selects:30 objects:30000	selects:30 objects:100000
db4o/5.7.001	363	901	1602	2974
db4o/6.0.001	640	1221	1928	3913
db4o	702	1280	2032	4032
db4o	691	1283	1997	4002

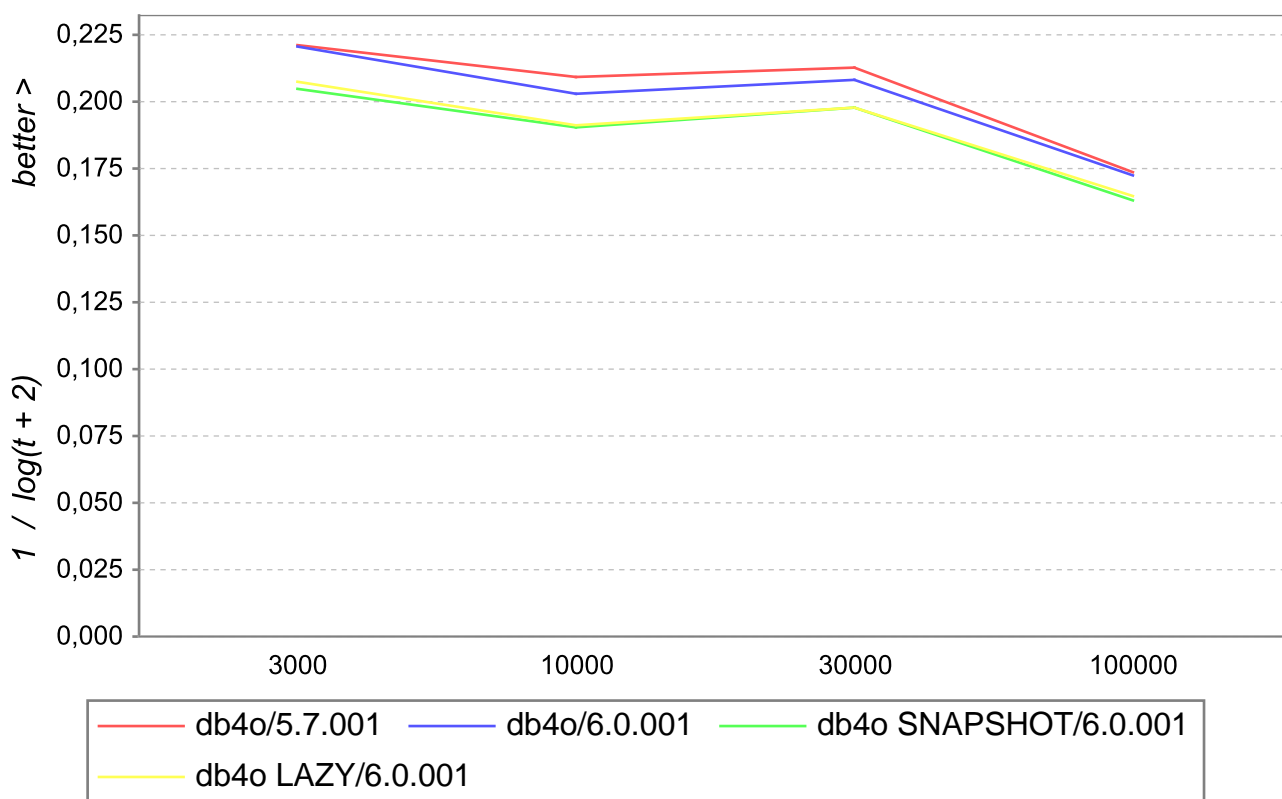


Circuit: Indianapolis

executes a variety of queries to test the efficiency of the query processor

Lap: queryOr

t [time in ms]	selects:30 objects:3000	selects:30 objects:10000	selects:30 objects:30000	selects:30 objects:100000
db4o/5.7.001	90	117	108	315
db4o/6.0.001	91	136	120	328
db4o	130	189	155	458
db4o	122	185	155	431

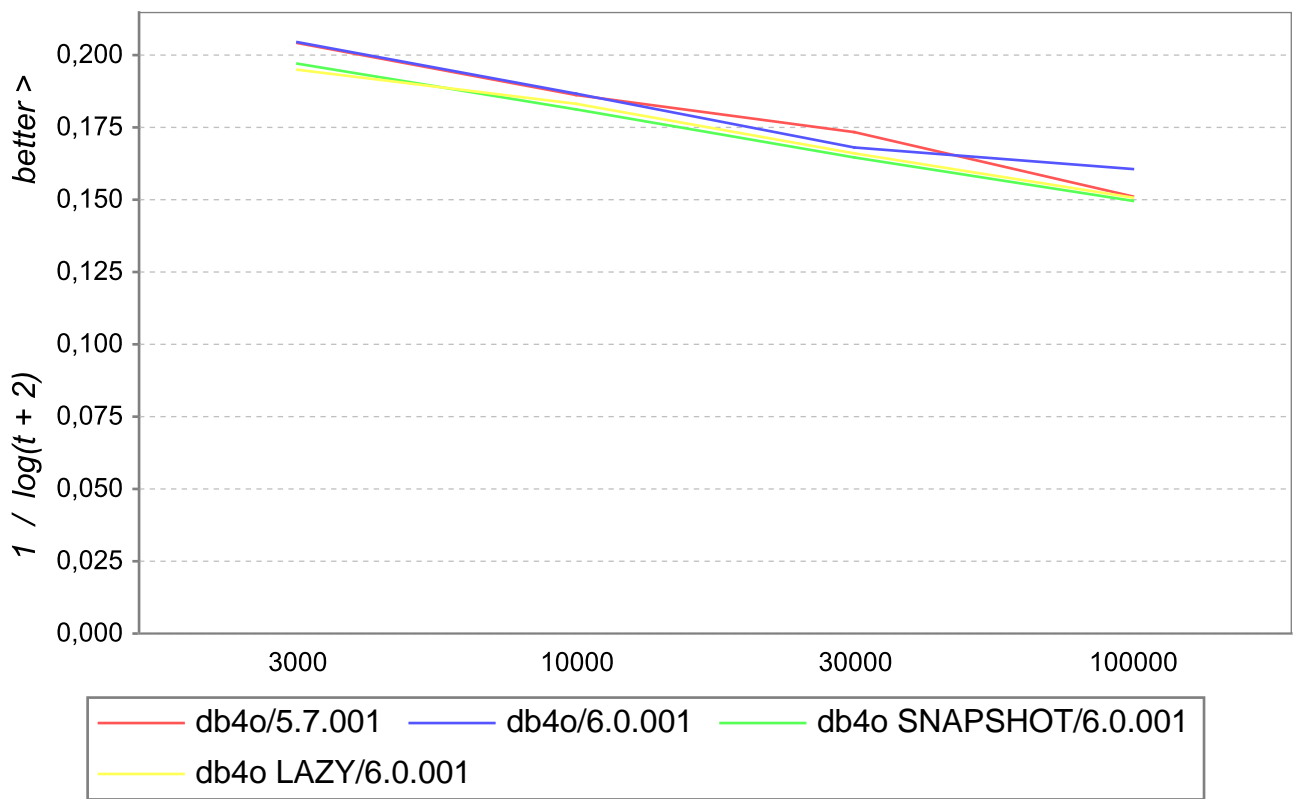


Circuit: Indianapolis

executes a variety of queries to test the efficiency of the query processor

Lap: queryOrRange

t [time in ms]	selects:30 objects:3000	selects:30 objects:10000	selects:30 objects:30000	selects:30 objects:100000
db4o/5.7.001	132	213	318	746
db4o/6.0.001	131	210	382	504
db4o	158	247	433	798
db4o	167	233	411	759

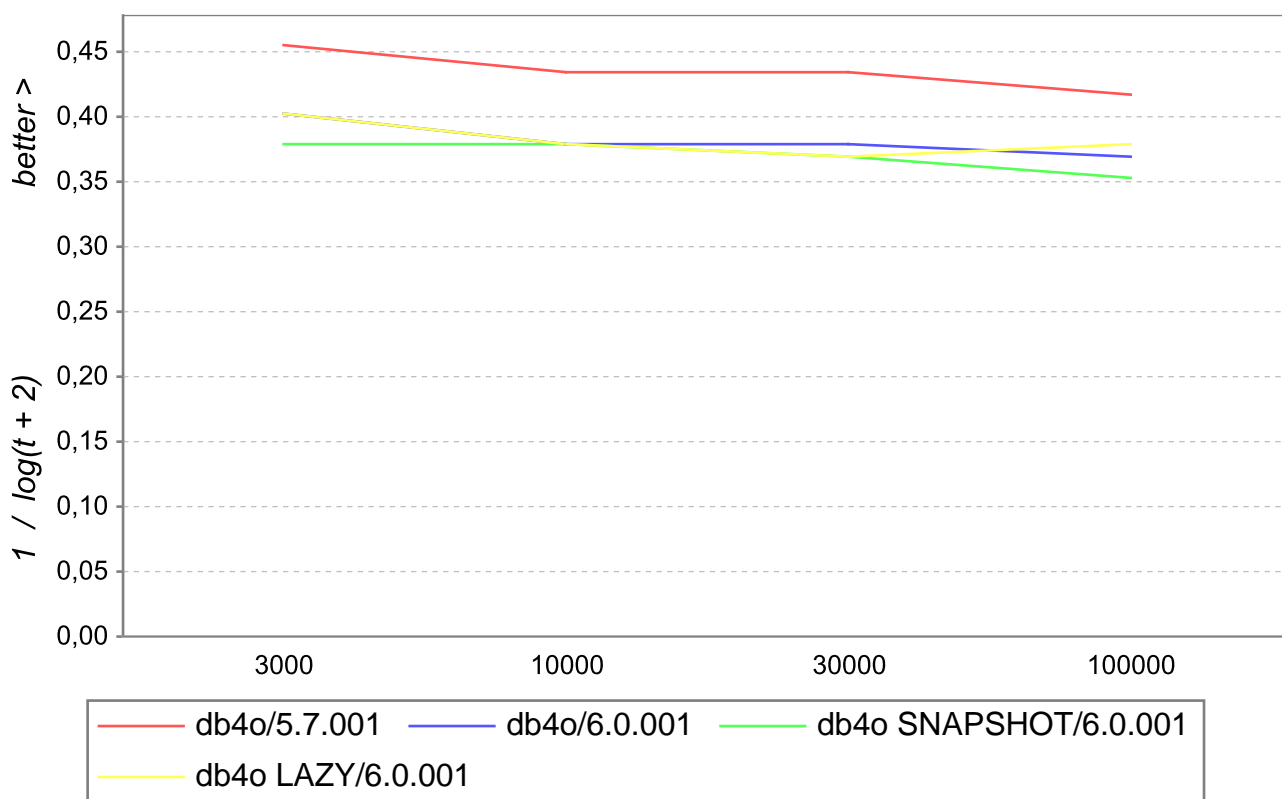


Circuit: Indianapolis

executes a variety of queries to test the efficiency of the query processor

Lap: queryNotGreater

t [time in ms]	selects:30 objects:3000	selects:30 objects:10000	selects:30 objects:30000	selects:30 objects:100000
db4o/5.7.001	7	8	8	9
db4o/6.0.001	10	12	12	13
db4o	12	12	13	15
db4o	10	12	13	12

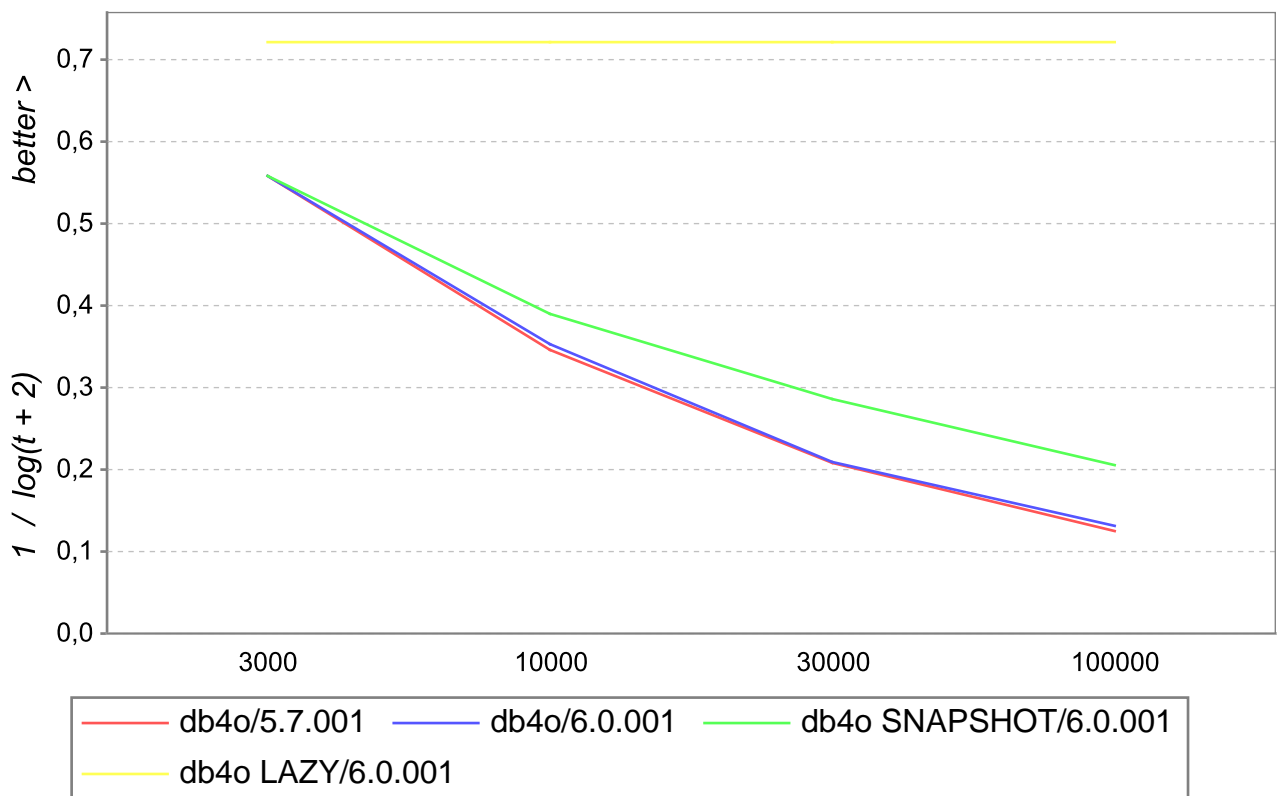


Circuit: Indianapolis

executes a variety of queries to test the efficiency of the query processor

Lap: `getSingleRandomObject`

t [time in ms]	selects:30 objects:3000	selects:30 objects:10000	selects:30 objects:30000	selects:30 objects:100000
db4o/5.7.001	4	16	120	2975
db4o/6.0.001	4	15	117	2021
db4o	4	11	31	128
db4o	2	2	2	2

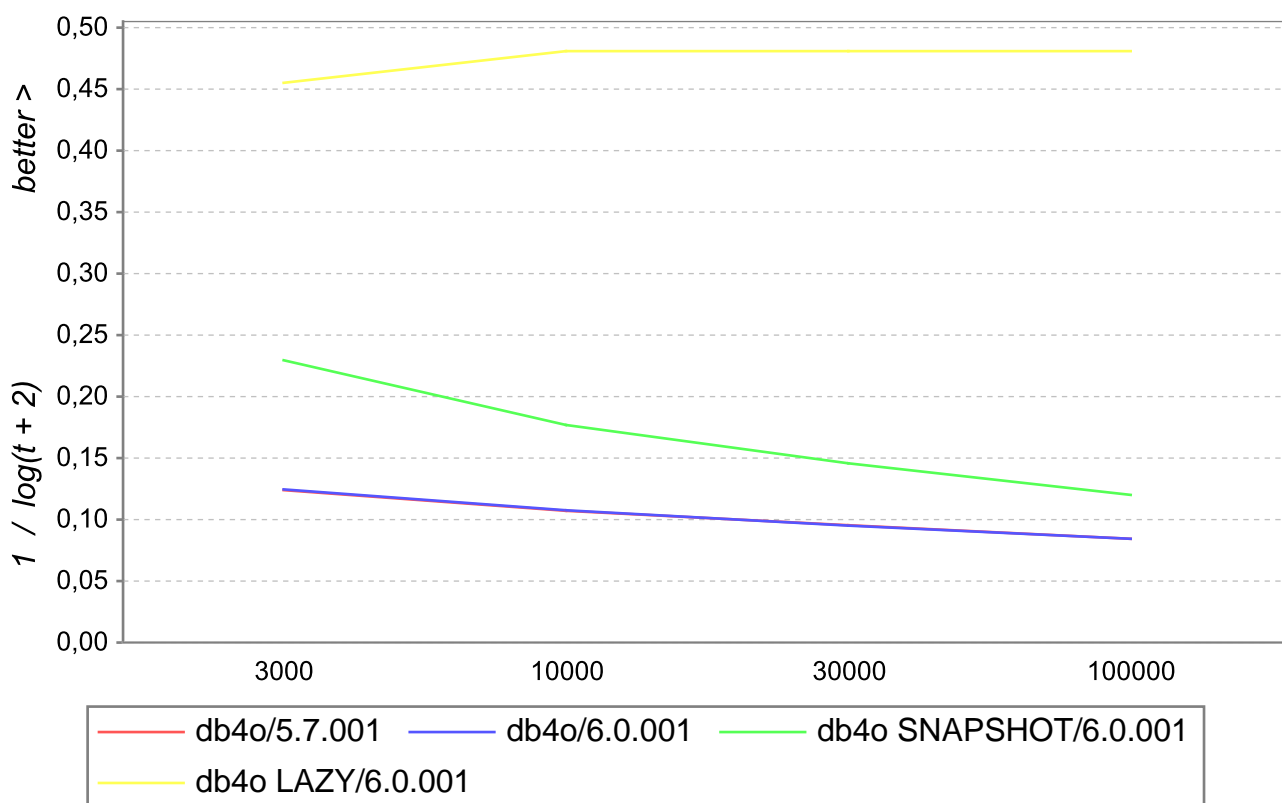


Circuit: Indianapolis

executes a variety of queries to test the efficiency of the query processor

Lap: getOneFromBigRangeQuery

t [time in ms]	selects:30 objects:3000	selects:30 objects:10000	selects:30 objects:30000	selects:30 objects:100000
db4o/5.7.001	3187	11319	35616	140166
db4o/6.0.001	3068	10885	36903	140474
db4o	76	283	952	4138
db4o	7	6	6	6

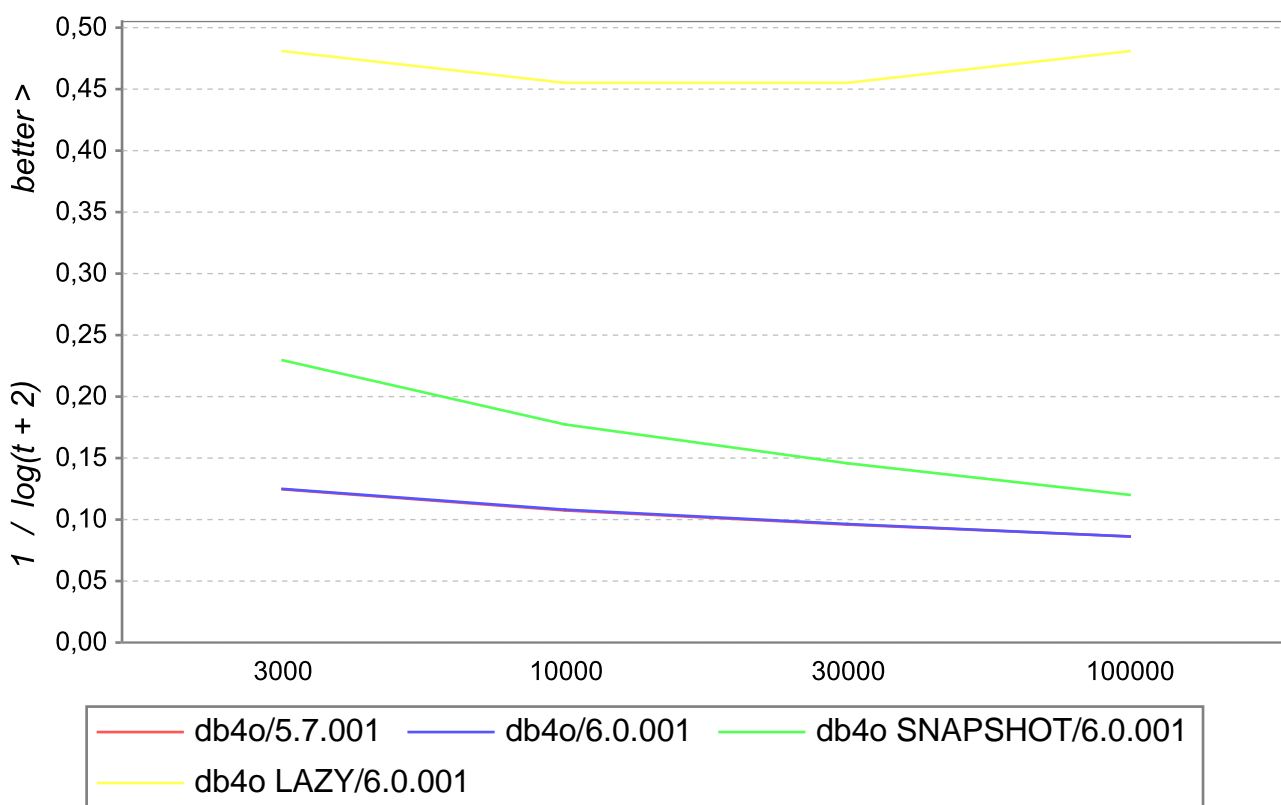


Circuit: Indianapolis

executes a variety of queries to test the efficiency of the query processor

Lap: getOneFromOrTwoLevelsQuery

t [time in ms]	selects:30 objects:3000	selects:30 objects:10000	selects:30 objects:30000	selects:30 objects:100000
db4o/5.7.001	3061	11035	33637	107551
db4o/6.0.001	2986	10422	32034	108629
db4o	76	279	950	4133
db4o	6	7	7	6



Circuit: Indianapolis

executes a variety of queries to test the efficiency of the query processor

Lap: addSingleObjectAndCommit

t [time in ms]	selects:30 objects:3000	selects:30 objects:10000	selects:30 objects:30000	selects:30 objects:100000
db4o/5.7.001	170	305	230	491
db4o/6.0.001	283	414	476	507
db4o	234	398	449	555
db4o	219	453	477	379

