Verity Technical Report: Additional Experimental Results

February 29, 2020

Abstract

This technical report is intended to serve as an additional resource to our paper: "Verity: Blockchains to detect insider attacks in DBMS". In this report we present the detailed experimental results for the performance evaluation of Verity. We refer the reader to our paper for detailed explanation of the framework and analysis of the results presented in this report.

1 SQL Queries

Q1: SELECT o_year, nation, (sum (volume) as mkt_share) from ((select (o_orderdate as o_year), ((l_extendedprice * (1 - l_discount)) as volume), (n2.n_name as nation) from part, supplier, lineitem, orders, customer, (nation as n1), (nation as n2), region where p_partkey = l_partkey and s_suppkey = l_suppkey and l_orderkey = o_orderkey and o_custkey = c_custkey and c_nationkey = n1.n_nationkey and n1.n_regionkey = r_regionkey and r_name = "asia" and s_nationkey = n2.n_nationkey and o_orderdate > '1995-01-01' and o_orderdate < '1996-12-31' and p_type = 'large plated tin') as all_nations);
Q2: SELECT * from region;
Q3: SELECT * from supplier;
Q4: SELECT * from nation;
Q5: UPDATE customer set c_name = "sjadfd", c_address = "kafawehrnj", c_phone="7894561265", c_acctbal = 22, c_mktsegment = "klasjfaw", c_comment="laksfnwe" where c_custkey = 91639739;
Q6: INSERT into nation (n_nationkey, n_name, n_regionkey, n_comment) values (93793619 ,"algeria", 123454556741 ,"haggle detect slyly agai");
Q7: INSERT into customer (c_custkey , c_name , c_address , c_nationkey , c_phone , c_acctbal ,c_mktsegment, c_comment) values (91639739 , "loren", "lipsum", 93793619 , "1234", 234, "muspil", "nerol");
Q8: INSERT into region (r_regionkey, r_name, r_comment) values (123454556741, "sambhal", "jhfasfhf kajh-fawerb idauhfwerbe aksfhnwejrb");
Q9: DELETE from nation where n_nationkey = 93793619;
Q10: DELETE from region where r_regionkey = 123454556741;
Q11: UPDATE supplier set s_nationkey = (select c_nationkey from customer where c_custkey= 91639739), s_phone = (select c_phone from customer where c_custkey= 91639739), s_comment ="askdenrjuhereu", s_acctbal = 2 + 10 where s_suppkey = 91639739 +1000;
Q12: SELECT s_acctbal, s_name, n_name, p_partkey, p_mfgr, s_address, s_phone, s_comment from part, supplier, partsupp, nation, region where p_partkey = ps_partkey and s_suppkey = ps_suppkey and p_size = 1 and s_nationkey = n_nationkey and n_regionkey = r_regionkey and r_name = "africa";
Q13: SELECT n_name, (sum(l_extendedprice * (1 - l_discount)) as revenue) from customer, orders, lineitem, supplier, nation, region where c_custkey = o_custkey and l_orderkey = o_orderkey and l_suppkey = s_suppkey and c_nationkey = s_nationkey and s_nationkey = n_nationkey and n_regionkey = r_regionkey and r_name = "asia" and o_orderdate >= "1995-03-09" and o_orderdate < "1996-03-09";
Q14: SELECT supp_nation, cust_nation, l_shipdate, (sum (volume) as revenue) from ((select (n1.n_name as supp_nation), (n2.n_name as cust_nation), l_shipdate, ((l_extendedprice * (1 - l_discount)) as volume) from supplier, lineitem, orders, customer, (nation as n1), (nation as n2) where s_suppkey = l_suppkey and o_orderkey = l_orderkey and c_custkey = o_custkey and s_nationkey = n1.n_nationkey and c_nationkey = n2.n_nationkey and ((n1.n_name = "india" and n2.n_name = "united states") or (n1.n_name = "united states" and n2.n_name = "india")) and l_shipdate > '1995-01-01' and l_shipdate < '1996-12-31') as shipping);
Q15: SELECT (sum(l_extendedprice * l_discount) as revenue) from lineitem where l_shipdate >= "1994-04-15" and l_shipdate < "1995-04-15" and l_discount > 0.04 - 0.01 and l_discount < 0.04 + 0.01 and l_quantity < 20 ;
Q16: SELECT o_orderpriority, (count(*) as order_count) from orders where o_orderdate >= "1995-03-09" and o_orderdate < "1995-06-09";
Q17: SELECT l_shipmode, o_orderpriority from orders, lineitem where o_orderkey = l_orderkey and (l_shipmode = "ship" or l_shipmode = "air") and l_commitdate < l_receiptdate and l_shipdate < l_commitdate and l_receiptdate >= "1995-01-01" and l_receiptdate < "1996-01-01";

Q18: INSERT into customer (c_custkey , c_name , c_address , c_nationkey , c_phone , c_acctbal ,c_mktsegment, c_comment) values (91639738 , “sumpil”, “renol”, 93793619 , “9242”, 234, “pilsum”, “rolen”) , (91639737 , “abc”, “def”, 93793619 , “1234”, 234, “yhbdsra”, “afgsdf”), (96244913, “pkjhbc”, “mnhgre”, 93793619 , “9543”, 234, “qaxcvf”, “iomnbgf”), (96244914, “yuthgbvfg”, “qgytrevd”, 93793619 , “75345”, 234, “liyhvdrt”, “qfgkdyv”), (96244915, “ramnabfubt”,
“njhiyfcvh”, 93793619 , “126789”, 234, “summinhsve”,
“qgjorutbbs”);
Q19: UPDATE customer set c_name=“ashdehhrbeki” where c_name = “sjadfd”;
Q20: UPDATE customer set c_name = “sjadfd”, c_address = “asndkwewhrwoer”, c_phone = “3245456458”, c_acctbal=22 ,c_mktsegment = “ajshuejre”, c_comment = “asjhdeke” where c_nationkey = 93793619;
Q21: DELETE from customer where c_nationkey = 93793619;
Q22: INSERT into supplier (s_suppkey, s_name, s_address, s_nationkey, s_phone, s_acctbal, s_comment) select (c_custkey + 1000), c_name, c_address, c_nationkey, c_phone, c_acctbal, c_comment from customer where c_nationkey = 93793619;
Q23: SELECT l_orderkey, (sum (l_extendedprice) as revenue), o_orderdate, o_shippriority from customer, orders, lineitem where c_mktsegment = “automobile” and c_custkey = o_custkey and l_orderkey = o_orderkey and o_orderdate < “1995-03-09” and l_shipdate > “1995-03-09”;
Q24: SELECT (sum (l_extendedprice * (1 - l_discount)) as promo_revenue) from lineitem, part where l_partkey = p_partkey and l_shipdate >= “1995-01-01” and l_shipdate < “1995-02-01”;
Q25: SELECT * from customer;
Q26: DELETE from supplier where s_nationkey = 93793619;
Q27: SELECT p_brand, p_type, p_size, (count (ps_suppkey) as supplier_cnt) from partsupp, part where p_partkey = ps_partkey and p_brand <> ‘brand#34’ and p_type not like ‘medium brushed brass’ and (p_size = 22 or p_size = 47 or p_size = 30 or p_size = 29 or p_size = 11 or p_size = 37 or p_size = 42 or p_size = 34 or p_size = 40);
Q28: SELECT * from part;
Q29: SELECT ps_partkey, (sum(ps_supplycost * ps_availqty) as value) from partsupp, supplier, nation where ps_suppkey = s_suppkey and s_nationkey = n_nationkey and n_name = “india”;
Q30: SELECT c_custkey, c_name, (sum (l_extendedprice * (1 - l_discount)) as revenue), c_acctbal, n_name, c_address, c_phone, c_comment from customer, orders, lineitem, nation where c_custkey = o_custkey and l_orderkey = o_orderkey and o_orderdate >= “1995-01-01” and o_orderdate < “1995-04-01” and l_returnflag = ‘r’ and c_nationkey = n_nationkey;
Q31: UPDATE supplier set s_acctbal = (select sum(c_custkey) from (customer as c), (nation as n) where c.c_nationkey = n.n_nationkey), s_phone = (select c_phone from customer where c_custkey= 91639738), s_comment = “asdherbejhrbeh” where s_suppkey = 100001;
Q32: SELECT s_suppkey, n_name, s_name from supplier, nation where supplier.s_nationkey = nation.n_nationkey;
Q33: SELECT s_suppkey, n_name, s_name from ((select * from supplier) as sup), nation where sup.s_nationkey = nation.n_nationkey;
Q34: SELECT s_suppkey, s_name, s_address, s_phone, total_revenue from supplier, ((select (l_suppkey), (max (l_extendedprice * (1 - l_discount)) as total_revenue) from lineitem where l_shipdate >= “1995-01-01” and l_shipdate < “1995-04-01”) as sup) where s_suppkey = sup.l_suppkey;
Q35: SELECT sn.s_name, rn.r_name from ((select n_name, s_name from (supplier as sup), nation where sup.s_nationkey = nation.n_nationkey) as sn), ((select n_name, r_name from (region as reg), nation where reg.r_regionkey = nation.n_regionkey) as rn) where sn.n_name = rn.n_name;
Q36: SELECT * from partsupp;
Q37: SELECT * from orders;
Q38: SELECT nation, o_year, (sum (amount) as sum_profit) from ((select (n_name as nation), (o_orderdate as o_year), (((l_extendedprice * (1 - l_discount)) - (ps_supplycost * l_quantity)) as amount) from part, supplier, lineitem, partsupp, orders, nation where s_suppkey = l_suppkey and ps_suppkey = l_suppkey and ps_partkey = l_partkey and p_partkey = l_partkey and o_orderkey = l_orderkey and s_nationkey = n_nationkey and p_name like ‘%ab%’) as profit);
Q39: SELECT l_returnflag, l_linestatus, (sum (l_quantity) as sum_qty), (sum (l_extendedprice) as sum_base_price), (avg (l_quantity) as avg_qty), (avg (l_extendedprice) as avg_price), (avg (l_discount) as avg_disc) from lineitem where l_quantity <= 20;
Q40: SELECT c_orders.c_custkey, (count(*) as custdist) from ((select c_custkey, o_orderkey from customer, orders where c_custkey = o_custkey and o_comment not like ‘%fi%al%’) as c_orders); select * from customer;
Q41: DELETE from supplier where s_suppkey > 20000;
Q42: INSERT into supplier (s_suppkey, s_name, s_address, s_nationkey, s_phone, s_acctbal, s_comment) select (c_custkey + 100000), n_name, c_address, c_nationkey, c_phone, c_acctbal, c_comment from (customer as c), (nation as n) where c.c_nationkey = n.n_nationkey ;
Q43: SELECT * from lineitem;

2 Runtime Statistics of Verity on SQL Queries

Table 1: Statistics for TPC-H Scaling Factor=0.001

Query No.	*Query Type*	*Is Nested Query?*	*Num Tables*	*Num tuples*	*Parser Time*	*DB Time*	*Blockchain Time*	*Total Time*	*Total time per tuple*
1	S	✓	7	0	0.0048	0.0101	0.0000	0.0149	-
2	S	✗	1	5	0.0000	0.0002	0.5198	0.5200	0.10
3	S	✗	1	10	0.0000	0.0003	0.9553	0.9556	0.10
4	S	✗	1	25	0.0000	0.0004	2.4439	2.4443	0.10
5	U	✗	1	1	0.0008	0.0787	2.3677	2.4472	2.45
6	I	✗	1	1	0.0008	0.0549	2.4174	2.4730	2.47
7	I	✗	1	1	0.0008	0.0722	2.4190	2.4920	2.49
8	I	✗	1	1	0.0000	0.0569	2.4903	2.5472	2.55
9	D	✗	1	1	0.0000	0.0782	2.4660	2.5443	2.54
10	D	✗	1	1	0.0000	0.0545	2.4784	2.5329	2.53
11	U(S)	✓	3	3	0.0040	0.0689	2.6079	2.6808	0.89
12	S	✗	5	15	0.0024	0.0024	1.4877	1.4925	0.10
13	S	✗	6	0	0.0024	0.0074	0.0000	0.0098	-
14	S	✓	5	42	0.0040	0.0101	4.1746	4.1887	0.10
15	S	✗	1	33	0.0016	0.0044	3.3358	3.3417	0.10
16	S	✗	1	58	0.0008	0.0020	5.7078	5.7106	0.10
17	S	✗	2	58	0.0008	0.0055	5.8929	5.8992	0.10
18	I	✗	1	5	0.0048	0.0662	12.0515	12.1225	2.42
19	U	✗	1	6	0.0000	0.0797	14.1046	14.1844	2.36
20	U	✗	1	6	0.0016	0.0700	14.0755	14.1471	2.36
21	D	✗	1	6	0.0000	0.0615	14.6301	14.6916	2.45
22	I(S)	✓	2	12	0.0016	0.0968	15.0597	15.1581	1.26
23	S	✗	3	111	0.0056	0.0100	11.0798	11.0954	0.10
24	S	✗	2	142	0.0024	0.0077	13.9107	13.9207	0.10
25	S	✗	1	150	0.0056	0.0019	14.5149	14.5224	0.10
26	D	✗	1	12	0.0024	0.0790	29.2906	29.3720	2.45
27	S	✗	2	168	0.0048	0.0020	16.4539	16.4607	0.10
28	S	✗	1	200	0.0048	0.0025	19.2345	19.2417	0.10
29	S	✗	3	0	0.0016	0.0013	0.0000	0.0029	-
30	S	✗	4	288	0.0080	0.0062	28.6726	28.6868	0.10
31	U(S)	✓	4	314	0.0032	0.0734	35.1650	35.2416	0.11
32	S	✗	2	344	0.0128	0.0016	35.1016	35.1160	0.10
33	S	✗	2	344	0.0088	0.0017	35.0040	35.0145	0.10
34	S	✓	2	416	0.0144	0.0074	41.3524	41.3743	0.10
35	S	✗	3	760	0.0168	0.0041	76.4268	76.4476	0.10
36	S	✗	1	700	0.0192	0.0066	67.3869	67.4127	0.10
37	S	✗	1	1500	0.0328	0.0152	143.5823	143.6303	0.10
38	S	✓	6	1266	0.0344	0.0158	126.0149	126.0651	0.10
39	S	✗	1	2406	0.0736	0.0248	241.0632	241.1616	0.10
40	S	✓	3	2504	0.0624	0.0158	248.9978	249.0760	0.10
41	D	✗	1	150	0.0008	0.0801	367.3438	367.4248	2.45
42	I(S)	✓	3	468	0.0152	0.1124	406.5742	406.7018	0.87
43	S	✗	1	6005	0.1432	0.0921	576.6234	576.8587	0.10

Table 2: Statistics for TPC-H Scaling Factor=0.002

Query No.	*Query Type*	*Is Nested Query?*	*Num Tables*	*Num tuples*	*Parser Time*	*DB Time*	*Blockchain Time*	*Total Time*	*Total time per tuple*
1	S	✓	7	0	0.0040	0.0302	0.0000	0.0342	-
2	S	✗	1	5	0.0000	0.0002	0.4767	0.4768	0.10
3	S	✗	1	20	0.0000	0.0004	1.6500	1.6505	0.08
4	S	✗	1	25	0.0000	0.0003	2.0154	2.0158	0.08
5	U	✗	1	1	0.0000	0.0808	2.3092	2.3901	2.39
6	I	✗	1	1	0.0008	0.0548	2.3493	2.4049	2.40
7	I	✗	1	1	0.0008	0.0535	2.3520	2.4063	2.41
8	I	✗	1	1	0.0008	0.0611	2.3856	2.4475	2.45
9	D	✗	1	1	0.0008	0.0674	2.3894	2.4576	2.46
10	D	✗	1	1	0.0000	0.0710	2.3958	2.4668	2.47
11	U(S)	✓	3	3	0.0032	0.0618	2.4882	2.5532	0.85
12	S	✗	5	40	0.0016	0.0039	3.2748	3.2804	0.08
13	S	✗	6	54	0.0016	0.0172	4.5378	4.5566	0.08
14	S	✓	5	66	0.0048	0.0204	5.4998	5.5250	0.08
15	S	✗	1	65	0.0048	0.0117	5.5543	5.5708	0.09
16	S	✗	1	113	0.0024	0.0029	9.4547	9.4600	0.08
17	S	✗	2	126	0.0024	0.0141	10.3405	10.3570	0.08
18	I	✗	1	5	0.0040	0.0842	11.7486	11.8367	2.37
19	U	✗	1	6	0.0016	0.0753	13.7905	13.8674	2.31
20	U	✗	1	6	0.0008	0.0703	13.8243	13.8954	2.32
21	D	✗	1	6	0.0000	0.0956	14.2985	14.3941	2.40
22	I(S)	✓	2	12	0.0000	0.0819	14.6229	14.7048	1.23
23	S	✗	3	189	0.0064	0.0230	15.7252	15.7546	0.08
24	S	✗	2	288	0.0080	0.0194	23.9018	23.9291	0.08
25	S	✗	1	300	0.0104	0.0035	24.7885	24.8024	0.08
26	D	✗	1	12	0.0008	0.0669	28.7344	28.8021	2.40
27	S	✗	2	374	0.0080	0.0062	30.9371	30.9514	0.08
28	S	✗	1	400	0.0104	0.0064	33.1052	33.1220	0.08
29	S	✗	3	450	0.0128	0.0055	37.2738	37.2922	0.08
30	S	✗	4	632	0.0136	0.0185	52.1661	52.1982	0.08
31	U(S)	✗	4	614	0.0168	0.0768	53.3380	53.4316	0.09
32	S	✗	2	664	0.0144	0.0038	55.0981	55.1163	0.08
33	S	✗	2	664	0.0216	0.0042	55.3800	55.4058	0.08
34	S	✓	2	884	0.0176	0.0153	73.0726	73.1055	0.08
35	S	✗	3	1416	0.0376	0.0058	117.6380	117.6813	0.08
36	S	✗	1	1500	0.0456	0.0113	123.8049	123.8617	0.08
37	S	✗	1	3000	0.0720	0.0333	247.1931	247.2984	0.08
38	S	✓	6	3924	0.0920	0.0411	324.6050	324.7381	0.08
39	S	✗	1	4730	0.1416	0.0469	390.4591	390.6476	0.08
40	S	✓	3	5040	0.1352	0.0331	416.6727	416.8410	0.08
41	D	✗	1	300	0.0016	0.0761	715.0222	715.0999	2.38
42	I(S)	✓	3	918	0.0312	0.0706	768.6587	768.7606	0.84
43	S	✗	1	11957	0.3472	0.1881	988.8332	989.3684	0.08

Table 3: Statistics for TPC-H Scaling Factor=0.005

Query No.	*Query Type*	*Is Nested Query?*	*Num Tables*	*Num tuples*	*Parser Time*	*DB Time*	*Blockchain Time*	*Total Time*	*Total time per tuple*
1	S	✓	7	168	0.0064	0.0739	13.7367	13.8171	0.08
2	S	✗	1	5	0.0000	0.0002	0.4544	0.4545	0.09
3	S	✗	1	50	0.0008	0.0007	4.2600	4.2614	0.09
4	S	✗	1	25	0.0000	0.0006	2.1579	2.1585	0.09
5	U	✗	1	1	0.0016	0.0725	2.3424	2.4164	2.42
6	I	✗	1	1	0.0008	0.1076	2.3547	2.4632	2.46
7	I	✗	1	1	0.0024	0.1110	2.3862	2.4997	2.50
8	I	✗	1	1	0.0016	0.1505	2.4079	2.5600	2.56
9	D	✗	1	1	0.0000	0.1152	2.4143	2.5295	2.53
10	D	✗	1	1	0.0000	0.1941	2.3970	2.5911	2.59
11	U(S)	✓	3	3	0.0016	0.0718	2.5159	2.5894	0.86
12	S	✗	5	65	0.0040	0.0066	5.4095	5.4200	0.08
13	S	✗	6	276	0.0056	0.0572	22.3793	22.4421	0.08
14	S	✓	5	252	0.0080	0.0741	20.7544	20.8365	0.08
15	S	✗	1	149	0.0016	0.0440	12.2877	12.3333	0.08
16	S	✗	1	275	0.0080	0.0092	22.5769	22.5941	0.08
17	S	✗	2	302	0.0104	0.0460	24.6150	24.6714	0.08
18	I	✗	1	5	0.0048	0.0934	11.8735	11.9716	2.39
19	U	✗	1	6	0.0016	0.0616	13.9002	13.9635	2.33
20	U	✗	1	6	0.0008	0.0633	14.0037	14.0678	2.34
21	D	✗	1	6	0.0008	0.0572	14.4794	14.5375	2.42
22	I(S)	✓	2	12	0.0000	0.0827	14.7840	14.8667	1.24
23	S	✗	3	510	0.0120	0.0555	41.6116	41.6790	0.08
24	S	✗	2	690	0.0192	0.0492	56.9924	57.0609	0.08
25	S	✗	1	750	0.0208	0.0077	62.1397	62.1681	0.08
26	D	✗	1	12	0.0000	0.0583	29.0842	29.1425	2.43
27	S	✗	2	1224	0.0272	0.0144	101.1121	101.1537	0.08
28	S	✗	1	1000	0.0208	0.0132	84.1594	84.1934	0.08
29	S	✗	3	702	0.0192	0.0110	57.7728	57.8030	0.08
30	S	✗	4	1740	0.0400	0.0547	142.9865	143.0812	0.08
31	U(S)	✓	4	1514	0.0336	0.0883	130.7288	130.8507	0.09
32	S	✗	2	1624	0.0256	0.0083	137.2414	137.2754	0.08
33	S	✗	2	1624	0.0392	0.0066	137.2102	137.2560	0.08
34	S	✓	2	2086	0.0512	0.0519	172.0651	172.1682	0.08
35	S	✗	3	3404	0.0920	0.0117	287.7352	287.8389	0.08
36	S	✗	1	3900	0.0984	0.0286	325.9065	326.0335	0.08
37	S	✗	1	7500	0.1632	0.0758	629.3559	629.5948	0.08
38	S	✓	6	10284	0.1912	0.1266	845.1213	845.4391	0.08
39	S	✗	1	12024	0.3024	0.1231	998.4503	998.8758	0.08
40	S	✓	3	12632	0.3104	0.0716	1,036.9132	1,037.2953	0.08
41	D	✗	1	750	0.0000	0.1782	1,805.9168	1,806.0949	2.41
42	I(S)	✓	3	2268	0.0640	0.1009	1,920.5538	1,920.7187	0.85
43	S	✗	1	30201	0.7736	0.4416	2,545.3044	2,546.5196	0.08

Table 4: Statistics for TPC-H Scaling Factor=0.01

Query No.	*Query Type*	*Is Nested Query?*	*Num Tables*	*Num tuples*	*Parser Time*	*DB Time*	*Blockchain Time*	*Total Time*	*Total time per tuple*
1	S	✓	7	232	0.0056	0.1506	19.7123	19.8685	0.09
2	S	✗	1	5	0.0000	0.0002	0.5194	0.5196	0.10
3	S	✗	1	100	0.0016	0.0012	8.6209	8.6237	0.09
4	S	✗	1	25	0.0000	0.0005	2.1427	2.1432	0.09
5	U	✗	1	1	0.0000	0.0500	2.4098	2.4598	2.46
6	I	✗	1	1	0.0000	0.0462	2.3785	2.4247	2.42
7	I	✗	1	1	0.0016	0.0476	2.5143	2.5634	2.56
8	I	✗	1	1	0.0008	0.0517	2.6475	2.6999	2.70
9	D	✗	1	1	0.0008	0.0585	2.4311	2.4904	2.49
10	D	✗	1	1	0.0000	0.0450	2.4606	2.5056	2.51
11	U(S)	✓	3	3	0.0048	0.0545	2.6042	2.6635	0.89
12	S	✗	5	185	0.0040	0.0114	15.8775	15.8929	0.09
13	S	✗	6	504	0.0112	0.1167	42.4564	42.5842	0.08
14	S	✓	5	630	0.0208	0.1415	53.5475	53.7098	0.09
15	S	✗	1	314	0.0136	0.0893	26.5843	26.6872	0.08
16	S	✗	1	556	0.0168	0.0162	47.1865	47.2195	0.08
17	S	✗	2	562	0.0144	0.0867	47.5119	47.6131	0.08
18	I	✗	1	5	0.0056	0.0492	12.1982	12.2530	2.45
19	U	✗	1	6	0.0016	0.0609	14.0186	14.0811	2.35
20	U	✗	1	6	0.0008	0.0511	13.9064	13.9583	2.33
21	D	✗	1	6	0.0008	0.0627	14.4362	14.4997	2.42
22	I(S)	✓	2	12	0.0016	0.0580	14.8368	14.8964	1.24
23	S	✗	3	918	0.0264	0.1092	77.6668	77.8024	0.08
24	S	✗	2	1428	0.0392	0.0890	121.1017	121.2299	0.08
25	S	✗	1	1500	0.0368	0.0143	128.2105	128.2616	0.09
26	D	✗	1	12	0.0016	0.0629	29.2358	29.3003	2.44
27	S	✗	2	2688	0.0608	0.0306	227.4749	227.5663	0.08
28	S	✗	1	2000	0.0528	0.0243	171.5856	171.6627	0.09
29	S	✗	3	1200	0.0200	0.0203	103.0772	103.1175	0.09
30	S	✗	4	3328	0.0616	0.1000	281.6997	281.8613	0.08
31	U(S)	✓	4	3014	0.0800	0.0713	260.8157	260.9671	0.09
32	S	✗	2	3224	0.0800	0.0128	275.8760	275.9688	0.09
33	S	✗	2	3224	0.0592	0.0118	274.1745	274.2455	0.09
34	S	✓	2	4200	0.1144	0.1039	356.8365	357.0548	0.09
35	S	✗	3	6752	0.1400	0.0231	574.6790	574.8421	0.09
36	S	✗	1	8000	0.2264	0.0678	684.1501	684.4443	0.09
37	S	✗	1	15000	0.4008	0.1347	1,277.5337	1,278.0693	0.09
38	S	✓	6	20274	0.4064	0.2829	1,719.5267	1,720.2160	0.08
39	S	✗	1	23887	0.6200	0.2436	2,031.0139	2,031.8774	0.09
40	S	✓	3	25314	0.6288	0.1508	2,146.0609	2,146.8405	0.08
41	D	✗	1	1500	0.0008	0.0862	3,664.6589	3,664.7459	2.44
42	I(S)	✓	3	4518	0.1200	0.0967	3,906.6102	3,906.8269	0.86
43	S	✗	1	60175	1.5880	0.8557	5,203.0431	5,205.4868	0.09