Class	Functionality
TrieNodeStorage	Store the storage nodes in smart contract storage MPT
SnapshotStorage	Store the flat storage data for the smart contract of current world state
TxLookup	Store the transaction and receipt lookup metadata
TrieNodeAccount	Store the account nodes in world state MPT
SnapshotAccount	Store the flat account nodes of current world state
HeaderNumber	Track the block hash to block ID mapping
BloomBits	Store the Bloom filter bits for log searching
Code	Store the smart contract bytecode storage
SkeletonHeader	Store the block headers for skeleton sync
BlackHeader	Store the block header data (e.g., parent block, timestamp, gas limit)
BlockReceipts	Store the transaction receipts for each block
BlockBody	Store the block body data (e.g., transactions and uncle blocks)
StateID	Store the world state version identifier
BloomBitsIndex	Store the data table of a chain indexer to track its progress
Ethereum-genesis	Store the genesis state for the database
SnapshotJournal	Track the in-memory differential layers across system restarts within a snapshot journal
Ethereum-config	Store the Ethereum network configuration
LastStateID	Track the StateID of the latest stored world state
Unclean-shutdown	Store the list of local crashes
SnapshotGenerator	Track the snapshot generation marker across restarts
TrieJournal	Track the in-memory trie node layers across restarts
DatabaseVersion	Track the database schema version
LastBlock	Track the latest known full block's hash
SnapshotRoot	Track the hash of the last snapshot
SkeletonSyncStatus	Track the skeleton sync status across restarts
LastHeader	Track the latest known block header's hash
SnapshotRecovery	Track the snapshot recovery marker across restarts
TransactionIndexTail	Track the oldest block whose transactions have been indexed
LastFast	Track the latest known incomplete block's hash during fast sync

Table 1. Classes and corresponding functionalities of KV pairs appeared in Geth.

Class	# KV	Key size	Value size
TrieNodeStorage	1656.6 M	37.6±0.0001	70.3±0.003
SnapshotStorage	1222.3 M	65	12.5±0.0006
TxLookup	386.2 M	33	4
TrieNodeAccount	367.0 M	18.49±0.0001	115.7±0.006
SnapshotAccount	269.4 M	33	15.9±0.002
HeaderNumber	21.5 M	33	8
BloomBits	10.7 M	43	398.0±0.11
Code	1.47 M	33	6732.7±10.0
SkeletonHeader	0.55 M	9	609.7±0.02
BlackHeader	0.27 M	31.0±0.06	217.7±1.05
BlockReceipts	0.09 M	41	75910.7±346.5
BlockBody	0.09 M	41	79348.1±340.2
StateID	0.09 M	33	8
BloomBitsIndex	0.005 M	15.0±0.003	32.0±0.009
Ethereum-genesis	1	49	710909
SnapshotJournal	1	15	8369153
Ethereum-config	1	48	603
LastStateID	1	11	8
Unclean-shutdown	1	16	33
SnapshotGenerator	1	17	7
TrieJournal	1	11	352749130
DatabaseVersion	1	15	1
LastBlock	1	9	32
SnapshotRoot	1	12	32
SkeletonSyncStatus	1	18	146
LastHeader	1	10	32
SnapshotRecovery	1	16	8
TransactionIndexTail	1	20	8
LastFast	1	8	32

Table 2. Storage pattern analysis of each class. We present the number of KV pairs in the unit of one million (M), except those with only one KV pair. We also provide the average key and value sizes in bytes. For classes with variable key and value sizes, we include the 95% confidence intervals (under normal distribution) to capture the variability in size.

Appendix

Table 1 presents the 29 classes of KV pairs appearing in CacheTrace. Note that the skeleton mentioned in the table is one of the synchronization mechanisms in Geth. It prioritizes downloading block headers to quickly establish the chain's structure, then fills in block data as needed for efficiency. It sits between the fast sync and full sync mechanisms, where fast sync downloads only the recent state and verifies block headers from the genesis block, while full sync processes every block and transaction from the genesis block to rebuild the entire state.

Table 2 presents the storage pattern analysis of each class. Despite of the Code, BlockReceipts, BlockBody, Ethereum-genesis, SnapshotJournal, and TrieJournal, small KV pairs (less than 1 KiB) dominate the KV store (over 99.6%).

Tables 3 and 4 present the KV operation distributions in

Class	% of all operations	Writes (%)	Reads (%)	Scans (%)	Deletes(%)
TrieNodeStorage	38.5	59.4	35.7	-	4.87
SnapshotStorage	17.9	46.9	45.0	0.002	8.09
TxLookup	11.1	52.0	-	-	48.0
TrieNodeAccount	23.5	62.4	37.7	-	0.003
SnapshotAccount	7.48	72.1	27.9	0.000001	0.006
HeaderNumber	0.05	74.9	25.1	-	-
BloomBits	0.02	97.8	2.20	-	-
Code	0.41	12.8	87.2	-	-
SkeletonHeader	0.05	16.8	83.2	-	-
BlockHeader	0.62	16.9	60.6	5.63	16.9
BlockReceipts	0.11	32.1	35.8	-	32.1
BlockBody	0.14	24.2	51.6	-	24.2
StateID	0.07	50.0	-	-	50.0
BloomBitsIndex	0.002	1.10	98.9	-	-
LastStateID	0.03	0.01	99.9	-	-
Unclean-shutdown	0.00004	50.0	50.0	-	-
LastBlock	0.04	99.7	0.28	-	-
SnapshotGenerator	0.0004	100.0	-	-	-
SnapshotRoot	0.0007	50.0	-	-	50.0
SkeletonSyncStatus	0.009	99.8	0.19	-	-
LastHeader	0.003	100.0	-	-	-
TransactionIndexTail	0.00009	59.9	40.1	-	-
LastFast	0.003	100.0	-	-	-

Table 3. Statistics of KV operation distribution in CacheTrace. We show the percentages of writes, reads, scans, and deletes in each class.

Class	% of all operations	Writes (%)	Reads (%)	Scans (%)	Deletes(%)
TrieNodeStorage	57.3	38.7	60.2	-	1.10
TxLookup	3.46	52.0	-	-	48.0
TrieNodeAccount	38.6	58.7	41.3	-	0.0005
HeaderNumber	0.03	41.3	58.7	-	-
BloomBits	0.006	94.3	5.74	-	-
Code	0.13	12.8	87.2	-	-
SkeletonHeader	0.05	6.02	75.6	-	18.4
BlockHeader	0.20	16.4	61.7	5.47	16.4
BlockReceipts	0.03	32.1	35.8	-	32.1
BlockBody	0.05	23.2	53.5	-	23.2
StateID	0.01	50.0	-	-	50.0
BloomBitsIndex	0.002	0.30	99.7	-	-
LastStateID	0.03	33.3	66.7	-	-
Unclean-shutdown	0.00002	50.0	50.0	-	-
LastBlock	0.01	98.9	1.05	-	-
SkeletonSyncStatus	0.002	99.2	0.75	-	-
LastHeader	0.01	100.0	-	-	-
TransactionIndexTail	0.00003	55.4	44.6	-	-
LastFast	0.01	100.0	-	-	-

Table 4. Statistics of KV operation distribution in BareTrace. We show the percentages of writes, reads, scans, and deletes in each class.

both CacheTrace and BareTrace of each class, respectively. We omit the classes used for system maintenance in the tables as they only have one read and write operation during the system's startup and shutdown, respectively. Since the CacheTrace is collected with the caching and snapshot acceleration mechanisms enabled, KV operations to the KV pairs belonging to the classes SnapshotStorage, SnapshotAccount, SnapshotRoot, SnapshotGenerator are only shown in Table 3.