

FoundationDB C client uses a custom fast-allocator to allocate memory from Operating System in large blocks, and then reuses this memory for various operations within the process. It is important to note that memory once allocated is never returned back to OS.

A major chunk of allocated memory is used by ongoing transactions to cache the keys/values read and written over their lifetime of the transaction. Memory used by transactions is returned back to allocator-pool when transaction is closed, and hence, it is crucial to close transactions in a timely manner. Most bindings take care of timely closing transactions as part of retry-loop.

Memory size will also depend on number of concurrently outstanding transaction - so, limiting those is another way to limit memory footprint of this component.

For read-only transactions, one can **disable read-your-writes** <<https://apple.github.io/foundationdb/javadoc/>> cache to limit transaction memory .

Important: For read-only transactions this does not change the semantics, but will involve making a network request to Storage Server should a key be read twice. For **read-write** transactions, this will change the semantics of transactions.

One can monitor memory utilization by the C client by enabling client side tracing and then looking for logs related to MemoryMetrics.