

Program performance can be dynamically improved by optimizing its frequent execution traces. Once traces are collected, they can be analyzed and optimized based on the dynamic information derived from the program's previous runs. The ability to record traces is thus central to any dynamic binary translation system. Recording traces, as well as loading them for use in different runs, requires code replication in order to represent the trace. This paper presents a novel technique which records execution traces by using an automaton called TEA (Trace Execution Automata). Contrary to other approaches, TEA stores traces implicitly, without the need to replicate execution code. TEA can also be used to simulate the trace execution in a separate environment, to store profile information about the generated traces, as well to instrument optimized versions of the traces. In our experiments, we showed that TEA decreases memory needs to represent the traces (nearly 80% savings).