

Dynamic Binary Translation (DBT) techniques have been largely used in the migration of legacy code and in the transparent execution of programs across different architectures. They have also been used in dynamic optimizing compilers, to collect runtime information so as to improve code quality. In many cases, DBT translation mechanism misses important low-level mapping opportunities available at the source/target ISAs. Hot code performance has been shown to be central to the overall program performance, as different instruction mappings can account for high performance gains. Hence, DBT techniques that provide efficient instruction mapping at the ISA level has the potential to considerably improve performance. This paper proposes ISAMAP, a flexible instruction mapping driven by dynamic binary translation. Its mapping mechanism, provides a fast translation between ISAs, under an easy-to-use description. At its current state, ISAMAP is capable of translating 32-bit PowerPC code to 32-bit x86 and to perform local optimizations on the resulting x86 code. Our experimental results show that ISAMAP is capable of executing PowerPC code on an x86 host faster than the processor emulator QEMU, achieving speedups of up to 3.16x for SPEC CPU2000 programs.