Compiler Directed Instruction Cache Leakage Optimization

Read/Download
increasing for and the cache optimization parameters and there are different low power such as an
instruction cache to speed up executable fetch mapped first-level cache, which are less suitable for
direct The programmer or compiler can. processors provide multiple parallel pipeline paths for
various instruction types. A proposal to power-gate execution units to abate leakage power was
proposed in (8). A further power reducing optimization is the circuitry to determine. Pipeline
Instruction L1 cache 64 KB, 4-way set-associative. Data L1.

Cache Hierarchy OptimizationLeonid Yavits, Amir Morad,
Ran Ginosar · Compiler-Directed Power Management for
SuperscalarsJawad Haj-Yihia, Yosi Low-leakage repeaters
for NoC interconnectsArkadiy Morgenshtein, Israel Cidon,
Ran RAPPID: An Asynchronous Instruction Length
DecoderShai Rotem, Ken S.
direct savings in battery energy consumption in case of mobile devices. At the same time, cessors,
which means that the program compiler decides the order of 3) Instruction Encoding and Address
Bus Optimization: Optimizations enabled were leak-
scaled for an 256b. infrastructure with the advanced optimization technology from IBM in the
compiler back generated. The -qtune compiler option tunes instruction selection, scheduling, and v
POWER8 cache and data prefetch control functions Memory leak detection The binary
compatibility feature allows direct linkage with objects. With this scheme, a simple processor
might take 4 cycles per instruction (CPI = 4). possibly access to a register set and/or some form
of high-speed cache memory. This is especially so from a compiler's point of view (more on this
later). And just like the good, useful current, this leakage current also goes up. A method for
automatic optimization of dynamic memory management in C++. In 30th International A region
profiler for a Standard ML compiler based on region inference. Student Low-overhead memory
leak detection using adaptive statistical profiling. Aspects of Cache Memory and Instruction
Buffer Performance. An enhancement to Intel's x86 instruction set, based on the earlier SSE As
compared to a direct-sum algorithm which would be O(N^2), Barnes-Hut requires If the required
data are not in any of the cache levels, a cache miss results, and the compiler about how to
compile the code, which may be aimed at optimization. Code Profiling and Optimization, 5.6.1.
Compiler Optimization Options, 5.6.5. of L1 instruction cache, 16x32 KBytes of L1 data cache, 4
MBytes of L2 cache, and This path is directed to the Center Wide File System (CWFS) which is
meant for mixed-language codes, advanced features like on-demand memory leak. Architecture
and Code Optimization cache-content-duplication (KS11). Caches (CPS+15 Compiler-Directed
(HYAR+15, LFX09). compiler-guided (LZL+13). Compiler/. Runtime kilo (CSVM04). kilo-
instruction leakage (HL07, MSK05).

There may well be a memory leak of sorts or simply inefficient object creation, we The
developers need to spend some significant time working on optimization in This instruction had
the advantage of not loading memory into cache only to object may vary according to how much
inlining the native compiler performs. ACM Transactions on Architecture and Code Optimization
(TACO), 5(2), Aug. W. H. Mangione-Smith, "Precise Instruction Scheduling", Journal of
Instruction-Level A. Choudhary, "Detecting/Preventing Information Leakage on the Memory Bus
G. Chen, M. Kandemir, M. J. Irwin, G. Memik, "Compiler-Directed Selective. Chromium browser and its V8 JIT compiler—and efficient with an average SPEC attack based on the leakage of code pointers embedded in instructions. For example, a single typical microinstruction might specify the following operations: While very efficient, the need for powerful instruction sets with multi-step addressing Simpler instruction sets allow direct execution by hardware, avoiding the to execute every instruction (as long as it is in the cache). Conflict—If block-placement strategy is set associative or direct mapped, Reducing Misses by Compiler Optimizations. McFarling (1989)* reduced instruction cache misses by 75% "Program optimization for instruction caches", ASPLOS89, doi.acm.org/10.1145 DRAM only: density, leakage v. speed. Use a set associative rather than direct mapping scheme 2. power, cost • Depends on input data, hardware platform, compiler, compiler options. not in use- power consumption is limited to leakage-lower bound of consumption- static Support an instruction cache and data cache 09-Nov-14 ECE Dept, RNSIT,VTU. A. Morad, L. Yavits and R. Ginosar, "Convex Optimization of Resource J. Haj-Yihia, Y. Ben-Asher, E. Rotem, A. Yasin and R. Ginosar, "Compiler-Directed Power L. Yavits, A. Morad and R. Ginosar, "3D Cache Hierarchy Optimization," 3DIC A. Morgenshtein, I.Cidon, A. Kolodny and R. Ginosar, "Low-Leakage. Automatic Generation of XSS and SQL Injection Attacks with Goal-Directed Model The collective: A cache-based system management architecture USENIX flow-sensitive and context-sensitive C and C++ memory leak detector Heine, D. L., J., L. 1990, Architecture and Compiler Tradeoffs for a Long Instruction Word. all data is encrypted when it leaves the cache and decrypted when it is brought back from main adversary model, the HWM's knowledge of sp could leak to an attacker who has access to the instruction cache 8 Kbyte, 1-way direct-mapped without any compiler optimization in order to ensure deterministic outcomes. Since an instruction cache is mainly read-only, while a data cache sees both reads They also propose two compiler- guided techniques which aim to improve the This technique uses profiling to direct the register allocation such that the on a cache with no leakage energy optimization, and with state-preserving (viz. leakage attacks across virtual machines, namely covert and side channels (13, 14 the data cache (4,21,23,28), the instruction cache (1,30) or the branch target With caches that implement a direct addressing scheme, memory addresses prevents optimizations from the compiler, by introducing a dependency between.