一個用於影像壓縮應用的嵌入式系統快取記憶體設計

施嘉政、王欣平

E-mail: 9509668@mail.dyu.edu.tw

摘 要

在現代計算機的微架構設計中，記憶體最佳化設計人員需要動態指令層追蹤的資料供最佳化設計用，然而動態指令層追蹤方式會產生龐大的資料難以分析及處理。所以本文將針對大家熟之的JPEG影像壓縮程式為例，提出一個新的動態指令層側描方式及一個命名為Melting的演算法。 新的側描方式採用階段式分析，將傳統函式層側描方式與現代指令層側描兩者優點並用，以減少指令追蹤側描所產生的資料量，並使用SimpleScalar/ARM 4.0模擬器產生JPEG編碼器動態指令。擷取的動態指令再透過Melting演算法分析所有動態指令，將各個指令加以個別統計，並且找出最常出現的相鄰連續指令區塊，本文稱為最常出現的最長指令序列。 這段序列可供記憶體架構最佳化設計用，也可用於指令壓縮等設計。新的側描方式及演算法也可應用於其他的微架構設計。

關鍵詞 : 資料探勘 ; 快取設計 ; SimpleScalar ; JPEG ; ARM

目錄

參考文獻

[1] Gregory K. Wallace,“ The JPEG Still Picture Compression Standard ”, CACM, Vol.34, No.4, pp.31-44, 1991.

[2] ITU/CCITT, Recommendation T.81, Digital compression and coding of continuous-tone still images, September. 1992.

[3] K. Karuri, M. Faruque, S. Kraemer, R. Leupers, G. Ascheid, and H. Meyr,“ Fine-grained Application Source Code Profiling for ASIP Design ”, In 42nd Design Automation Conference, pp.329-334, June 2005 [4] T. Ball,“ Efficiently Counting Program Events with Support for on-line Queries ”, ACM Transactions on Programming Languages and Systems, September. 1994.

[5] T. Ball, J. R. Larus,“ Optimally Profiling and Tracing Programs ”, ACM Transactions on Programming Languages and Systems, Volume 16, Issue4, pp.1319-1360, July 1994.

[6] J. R. Larus,“ Whole Program Paths ”, Proceedings of the SIGPLAN 99 Conference on Programming Languages Design and Implementation(PLDI 99), May 1999, Atlanta Georgia.

[7] Erez Perelman, Trishul M. Chilimbi, Brad Calder, Variational Path Profiling, Proceeding of the International Conference on Parallel Architectures and Compilation Techniques(PACT), September. 2005.

[8] W.-C. Hsu, J. Lu, P.-C. Yew, D. Chen,“ Dynamic trace selection using performance monitoring hardware sampling ”, International Symposium on Code Generation and Optimization, pp.79-90, March 2003.

[9] B. Cmelik,“ SpixTools Introduction and User ’ s Manual ”, Technical Report SMLI TR-93-6, Sun Microsystems Laboratory, Mountain View, CA, February. 1993.

[10] A. Srivastava and A. Eustace,“ ATOM: A system for building customized program analysis tools ”, In ACM conference on Programming Language Design and Implementation, pp.196-205, Orlando, FL, June 1994.

[11] L. Benini, F. Menichelli, M. Olivieri,“ A class of code compression schemes for reducing power consumption in embedded microprocessor systems ”, IEEE Transactions on Computers, Volume 53, Issue 4, pp.467-482. April 2004.

[12] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, R. B. Brown. Mibench, A free,“ commercially representative embedded benchmark suite ”, In Proceedings of the IEEE 4th Annual Workshop on Workload Characterization, 2001.

[13] Mibench Benchmark, http://www.eecs.umich.edu/mibench/.

[14] SimpleScalar Version 4.0 , http://www.simplescalar.com/ [15] T.-C. Chiueh and P. Pradhan,“ Cache memory design for network processors“ , High-Performance Computer Architecture, pp.409-418, 2000.

[16] P. Stefan, K. Dhireesha, and J. Eugene,“ Cache performance of video computation workloads ”, Digital and Computational Video,

pp.169-175, 2002.
[17] Dinesh C. Suresh, Frank Vahid, Greg Stitt, Jason R. Villarreal, and Walid A. Najjar, “ Profiling tools for hardware/software partitioning of embedded applications. ” Proceedings of the 2003 ACM SIGPLAN conference on Language, compiler, and tool for embedded systems, pp.189-198, 2003.
[18] A. J. Smith, “ Cache memories ”, ACM Computing Surveys 14, No.3, pp.473-530, 1982.
[19] N. Linda and L. Jilia, “ The Essentials of Computer Organization and Architecture ”, Jones and Bartlett Publishers, Inc., 2003.
[20] D. A. Patterson and J. L. Hennessy, “ Computer Organization & Design ”, Second edition, Morgan Kaufmann Publishers, San Francisco.
[21] http://www.gnu.org/software/binutils/manual/gprof-2.9.1/html_mono/gprof.html [22] http://kprof.sourceforge.net/ [23] 楊智喬，Xtensa可組態處理器及其應用（下），國家晶片系統設計中心。