

Investigación sobre la arquitectura de computadoras

Arquitecturas modernas emergentes de RISC y CISC

La conversión de instrucciones entre arquitecturas distintas, como CISC y RISC, representa un desafío importante en la compatibilidad de los sistemas modernos.

Las arquitecturas CISC cuentan con sus instrucciones complejas que pueden realizar varias operaciones en una sola línea de código.

Por otro lado, las arquitecturas RISC utilizan instrucciones más simples y uniformes que las CISC, y estas son diseñadas para mejorar la eficiencia y velocidad de ejecución.

Al traducir un programa de CISC a RISC, cada instrucción compleja debe descomponerse en varias instrucciones más simples, es decir, generando lo que se conoce como una inflación de instrucciones.

Esta inflación es una fuente principal de sobrecarga en los traductores binarios dinámicos, pues una sola instrucción original puede convertirse en múltiples instrucciones en la arquitectura destino.

La arquitectura destino se refiere al tipo de arquitectura de procesador en la que se va a ejecutar el código después de que se haya traducido.

Aunque existen técnicas para reducir esta inflación, en la cual todavía se presenta una pérdida considerable de rendimiento en muchas herramientas actuales.

Para analizar y mejorar este proceso, se ha desarrollado una herramienta llamada Deflater que evalúa la inflación de instrucciones y facilita la optimización de traductores binarios.

Deflater funciona como una caja negra, es decir, que no requiere acceso al código interno del traductor sino que este observa solamente el resultado de la traducción.

La herramienta incluye un modelo matemático que calcula la inflación global, además de un simulador que permite observar el comportamiento del código traducido en tiempo real.

También cuenta con pruebas automatizadas que evalúan diferentes traductores sin necesidad de intervención manual.

Gracias a estas características Deflater ha mejorado el rendimiento de traductores como QEMU, logrando incrementos significativos tras aplicar sus recomendaciones.

Esto demuestra que, con un análisis detallado, es posible reducir la sobrecarga producida por la traducción entre arquitecturas, alcanzando una ejecución más eficiente en plataformas modernas.

<https://dl.acm.org/doi/pdf/10.1145/3640813>

Resumen de actividad | alu_documentos | Biblioteca UTEQ Koha | Scopus - Resultados de búsqueda | An Instruction Inflation Analyzing Framework for Dynamic Binary Translators | uteq - inicio

dl.acm.org/doi/pdf/10.1145/3640813

An Instruction Inflation Analyzing Framework for Dynamic Binary Translators

1 / 25 | 100% | [Icons]

China
FUXIN ZHANG, State Key Lab of Processors, Institute of Computing Technology, CAS, China and University of Chinese Academy of Sciences, China

Dynamic binary translators (DBTs) are widely used to migrate applications between different instruction set architectures (ISAs). Despite extensive research to improve DBT performance, noticeable overhead remains, preventing near-native performance, especially when translating from complex instruction set computer (CISC) to reduced instruction set computer (RISC). For computational workloads, the main overhead stems from translated code quality. Experimental data show that state-of-the-art DBT products have dynamic code inflation of at least 1.4x. This indicates that on average, more than 1.4x host instructions are needed to emulate one guest instruction. Worse, inflation closely correlates with translated code quality. However, the detailed sources of instruction inflation remain unclear.

To understand the sources of inflation, we present *Inflator*, an instruction inflation analysis framework comprising a mathematical model, a collection of black-box unit tests called *benchMIA*Co, and a trace-based simulator called *InflaSim*. The mathematical model calculates overall inflation based on the inflation of individual instructions and translation block optimizations. *benchMIA*Co extracts model parameters from DBTs without accessing DBT source code. *InflaSim* implements the model and uses the extracted parameters from

This is a new article, not an extension of a conference paper. The content presented in this article is intended solely for non-profit educational purposes and shall not be used for commercial applications.
This work was supported by the National Key Research and Development Program of China, project number 2022YFB3301004.
Authors' addresses: B. Xie, Y. Yan, C. Yan, X. Li, Y. Lan, X. Wu, and F. Zhang, State Key Lab of Processors, Institute of Computing Technology, CAS, 6 Kexueyuan Nantie, Zhongguancun, Haidian, Beijing, China, 100190 and University of Chinese Academy of Sciences, No. 1 Tangshan East Rd, Haidian, Beijing, China, 101400; e-mails: xiebenyi@ict.ac.cn, yanyc@ict.ac.cn, yanyc@ict.ac.cn, lixy@ict.ac.cn, lanx@ict.ac.cn, fuzhang@ict.ac.cn; S. Tao and Z. Zhang, University of Science and Technology of China, No. 96, Jiaozui Road Huobei District, Hefei, Anhui, China, 230026; e-mails: taosicheng@ustc.edu.cn, zhangzhuangzhuang@ustc.edu.cn; T. Liu, The University of Texas at San Antonio, 1 UTSA Circle, San Antonio, Texas, USA, 78249; e-mail: tianyi.liu@utsa.edu; T. Zhang, Loongson Technology Co. Ltd., ICT Loongson Park, Daotianqin Rd, Haidian, Beijing, China, 100095 and Institute of Computing Technology, CAS, 6 Kexueyuan Nantie, Zhongguancun, Haidian, Beijing, China, 100190; e-mail: zhangfuxin@loongson.cn.

This work is licensed under a Creative Commons Attribution-NonCommercial International 4.0 License.
© 2024 Copyright held by the owner/authors.
ACM 1544-3566/2024/03-ART28
https://doi.org/10.1145/3640813

ACM Trans. Arch. Code Optim., Vol. 21, No. 2, Article 28. Publication date: March 2024.

Aula Virtual | alu_documentos | Biblioteca UTEQ | Scopus - Documentos | An Instruction Inflation Analyzing Framework for Dynamic Binary Translators | uteq - inicio | sello de la uteq | INFORMACIÓN

scopus.com/record/display.uri?eid=2-s2.0-85194391315&origin=resultslist&sort=pdf-f&src=s&sort=b&sdt=b&us=TITLE-ABS-KEY%26risc+and+cisc%29&relpos=2

Brought to you by Universidad Técnica Estatal de Quevedo

Scopus Search Sources SciVal ? [Icons]

< Back to results | < Previous 3 of 345 Next >

Download Print Save to PDF Save to list Create bibliography

ACM Transactions on Architecture and Code Optimization • Open Access • Volume 21, Issue 2 • 23 March 2024 • Article number 28

An Instruction Inflation Analyzing Framework for Dynamic Binary Translators

Xie, Benyi^{a, b} ; Yan, Yue^{a, b} ; Yan, Chenghao^{a, b} ;
Tao, Sicheng^c ; Zhang, Zhuangzhuang^c ;
Li, Xinyu^{a, b} ; Lan, Yanzhi^{a, b} ; Wu, Xiang^{a, b} ;
Liu, Tianyi^d ; Zhang, Tingting^{a, f} ; Zhang, Fuxin^{a, b}

Save all to author list

Document type
Conference Paper • Gold Open Access

Source type
Journal

ISSN
15443566

DOI
10.1145/3640813

View more

Cited by 2 documents

Tiaozhuan: A General and Efficient Indirect Branch Optimization for Binary Translation

Li, X., Guo, G., Lan, Y.
(2025) ACM Transactions on Architecture and Code Optimization

Indirect Transfer Association AI Algorithm in Dynamic Translation System

Hu, Y.
(2025) International Conference on Intelligent Systems and Computational Networks, ICISCN 2025

View all 2 citing documents