

DOI: 10.7652/xjtuxb201306012

一种 ARM 存储模型的快速检测方法

吕正¹, 陈昊^{2,1}, 陈峰¹

(1. 西北大学信息科学与技术学院, 710069, 西安; 2. 中国劳动关系学院高职学院, 100048, 北京)

摘要: 针对 ARM 多核处理器存储模型正确性的快速检测问题,提出了一种利用时间序和悬空窗口的有界特性的快速检测方法,并实现了检测工具。该方法给出了 ARM 存储模型基于 barrier 的弱一致性模型的公理语义,通过定期扫描处理器的性能计数器获得访存指令操作间的时间约束关系。检测工具由随机指令发生模块、多核处理器性能计数器记录模块和结果分析模块 3 部分组成,它的低算法时间复杂度特性使其能够有效处理上百万行 ARM 访存指令程序。检测工具使用 C++ 语言实现,可以在运行时动态调整指令流的长度参数,具有很好的扩展性。利用支持 ARM MPCore 的模拟器进行了实验,并用手工的方法在指令流执行序列中注入了几个错误,以验证程序结果是否违反 ARM 存储模型。实验结果表明,检测工具能够正确发现上述注入错误,检测方法和检测工具可以有效检测 ARM 多核处理器存储模型的正确性。

关键词: ARM 处理器;存储模型;正确性检测

中图分类号: TP311 **文献标志码:** A **文章编号:** 0253-987X(2013)06-0068-05

A Novel Validation Method for ARM Memory Model

LÜ Zheng¹, CHEN Hao^{2,1}, CHEN Feng¹

(1. School of Information Science and Technology, Northwest University, Xi'an 710069, China;

2. Advanced Vocational School, China Institute of Industrial Relations, Beijing 100048, China)

Abstract: A low time complexity method for checking the correctness of ARM memory model is proposed by identifying the pending windows of microprocessor and introducing the restrictions of time order, and a validation tool is presented. The ARM memory model is explored and the formal definition of the model is given in the method. The time order constraints are obtained by scanning the performance counters of the ARM memory system periodically. The validation tool consists of three parts, namely, a random instruction generation module, a performance counter recording module and a results analysis module. The tool can effectively handle millions of lines of ARM multicore program because of the low time complexity of the algorithm. The validation tool is written in C++, and the length of the access memory instructions can be adjusted dynamically in run-time, which supports high extendability. Experiments with several infected errors in an ARM MPCore simulator show that the validation tool can detect these errors correctly, and that the proposed validation method and tool can effectively verify the ARM memory model.

Keywords: ARM processor; memory model; correctness validation

ARM(Advanced RISC Machine)是一种精简指令集(RISC)中央处理器架构。英国的 ARM 公司

目前已推出同属于 ARM 架构的 20 多种处理器,并已在许多嵌入式系统设计中广泛应用^[1]。其中,ARM 11 MPCore 和 ARM Cortex-A9 MPCore 是多核处理器版本。ARM 11 MPCore 最多支持 4 个计算核,这些核通过内置 SCU 实现高效一致性,可以高效支持 16~64 kB L1 的高速缓存,其系统结构如图 1 所示。

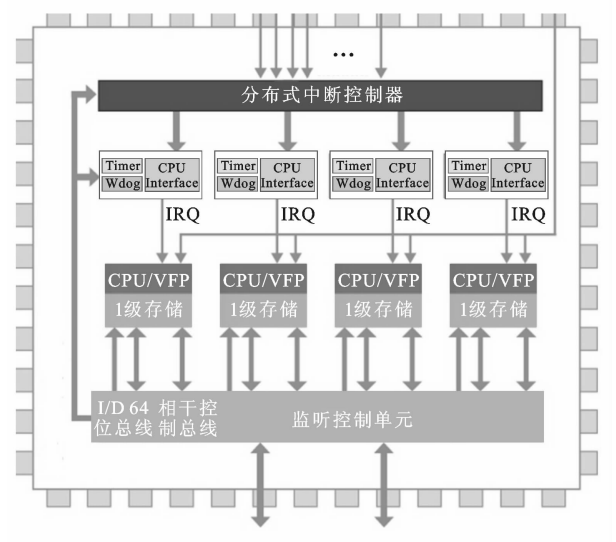


图 1 ARM MPCore 的系统结构图

存储一致性是共享存储系统的重要特性,它决定着并行程序执行结果的正确性。顺序一致性模型便于程序员编程,但不利于并行性的扩展。ARM 存储一致性模型是一种弱存储一致性模型,这种存储一致性模型对同一处理器核内的访存指令的约束很少,因此同一核内的各种访存指令可以乱序执行,能够很好地提高处理器的整体性能,同时又具备高扩展性。为了保证程序执行的正确性,ARM 处理器提供了多种同步指令,如 DMB (Data Memory Barrier) 和 DSB (Data Synchronization Barrier) 指令,程序员可以在程序中插入这些同步指令,获得所期望的顺序关系^[1]。

图 2 是一个使用 DMB 同步指令的 ARM 存储模型的并发程序执行实例,我们用 $S[A] \# V$ 表示对地址 A 写入值 V ,用 $L[A]=V$ 表示从地址 A 读入数据 V , P_1 、 P_2 和 P_3 是 3 个进程。假设 P_2 中存在一个 DMB 同步指令,那么根据 ARM 存储模型,不仅 P_2 中的 $L[A]=1$ 在 $L[B]=0$ 之前发生, P_1 中的 $S[A] \# 1$ 也在 P_3 中的 $S[B] \# 1$ 之前发生。

对 ARM 存储模型正确性的验证是一个相对困难的问题,目前的主流验证和检测工具均无法直接验证这种模型。我们首先给出 ARM 存储模型的形式

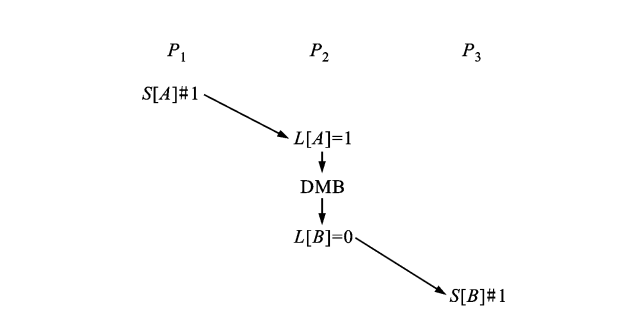


图 2 ARM 存储模型的程序执行

定义,在此基础上提出检测 ARM 存储模型的算法,并对 MOTEC 存储一致性验证工具进行扩展,以便能够检测 ARM 存储模型。

1 相关工作

对多处理器或多核处理器的存储一致性模型的验证和检测方法主要有 2 种:一种是对处理器的存储模型进行形式定义,然后根据模型的形式定义,有针对性地手工设计和生成小规模测试用例,根据测试用例的执行结果来检验待测处理器是否违反存储一致性模型,这种方法称为试金石测试 (litmus test),由 Collier 提出,后被用于 Alpha、Intel IA-64 等处理器的测试^[2-4];另一种重要的方法是随机生成具有冲突访问的并行程序,在并行程序运行后对程序的执行结果进行检测。Hangal 和 Manovit 等人提出了一个用于验证 Sun 的 SPARC 处理器 TSO 模型的存储一致性检测工具 TSOtool,并采用向量时钟来降低时间复杂度,使用回溯技术获得验证方法的完备性^[5-7]。Roy 等人提出了一种类似的存储一致性验证工具,用于验证 Intel 的多种处理器^[8]。总体来说,这些方法的缺点是算法时间复杂度高,无法验证规模很大的随机并发程序,而且验证的存储一致性模型比较有限。LCHECK 是一种线性时间复杂度的存储一致性验证工具,其算法的时间复杂度为 $O(p^3n)$ ^[9-10],但是该工具无法直接验证 ARM 存储模型。MOTEC 验证工具通过对处理器的性能计数器进行定期扫描,可以获得不同的访存操作集合间的时间约束关系,时间复杂度同样为 $O(p^3n)$ 。该工具能够用于流片后阶段的验证,具有更广泛的通用性^[11-12]。遗憾的是,这些工具都不能直接验证 ARM 存储模型。我们对存储一致性模型的序关系进行了扩充,可以在该框架下定义基于 barrier 的弱存储一致性模型,并在此基础上实现了一个验证工具原型,可以有效地检测 ARM 存储模型。

2 存储一致性模型的序关系

本文的存储一致性模型序关系与时间约束关系是对文献[9]的研究的扩充和改进。假设 ARM 并发程序包含读(r)和写(w)2种访存操作,以及 DMB 同步操作(本文暂不考虑 DSB 操作),除了已有的程序序和执行序外,还需要对存储一致性模型的序关系进行扩充。

(1)若操作 u 和操作 v 在同一进程中并访问同一单元, u 和 v 至少有一个操作是写操作,且 u 在程序中出现在 v 的前面,则称 u 在单机序上先于 v 。

为了形式化地表示 ARM 存储模型,还需给出如下定义:在满足 ARM 存储模型的并程序中,对任一进程中的 DMB 操作,所有在程序序上先于该 DMB 的访存操作指令序列构成一个同步程序块,记为 U_B ;所有在程序序上后于该 DMB 的访存操作的指令序列构成一个同步程序块,记为 L_B 。

(2)对于任意2个操作 $u_1 \in U_B, u_2 \in L_B$,均有 u_1 在同步序上先于 u_2 。

(3)若操作 u_1 在上述序的传递闭包关系上先于操作 u_2 ,则称 u_1 在全局序上先于 u_2 。

不同的存储一致性模型有着不同的全局序,程序执行结果满足某一存储一致性模型的正确性标准为:用该存储一致性模型定义的全局序导出并发程序执行结果的有向图,有向图中不存在环。可以通过 ARM 存储模型静态地确定单机序和同步序,但是 ARM 并发程序的执行序和全局序则需要根据 ARM 处理器的运行结果推导出来。

一条访存操作的进入时间是指从访存操作处理器的指令高速缓存中读出的时刻。处理器内部的部件(操作窗口、访存队列等)的大小和长度都是有限的,访存操作一定会在有限时间内完成并在一个时刻提交,这一时刻记为访存操作的提交时间。如果访存操作 u 的提交时间在访存操作 v 的进入时间之前,则称 u 在时间序上先于 v 。可以将全局序和时间序结合起来形成一种新的偏序关系,其实质是在原有存储一致性模型规定的序关系上增加了时间序关系,称之为时间全局序。这种序关系是各种序关系的传递闭包关系^[9-10]。

用时间全局序关系表示的程序执行结果图称为 TGO 图。根据文献[9]的时间序约束定理,可以通过 TGO 图是否有环来判断程序执行结果的正确性。

3 执行窗口与时间序

可以利用多核处理器的性能计数器获得访存操

作的进入和提交时间信息。如果知道某一时间点所有处理器核的性能计数,就可以得到访存操作集合的上界和下界信息。该类信息是受限的,每个访存操作的具体进入时间和提交时间是无法一一获得的,但如能周期性地扫描性能计数器,则每个扫描周期都能获得当前操作集合的上、下边界信息。由于 ARM 处理器没有专门用于记录访存操作集合上界和下界的性能计数器,所以用程序计数器来代替。由于处理器内部部件的有限性,可以根据程序计数器的值推算出访存操作集合的上界和下界。TGO 图被2个边界约束分成3块:未进入区域、活动区域和已提交区域。操作集合间的时间序关系可通过执行窗口这个概念得到,具体定义请参见文献[12],其主要结论是执行窗口导出的时间序定理。

定理 设 W_{A_i} 是一个活动窗口, W_{p_i} 是其对应的执行窗口。对任一操作 u^i ,若 u^i 发生在 W_{p_i} 的下界之前,则它一定在时间序上发生在 W_{A_i} 内的所有操作前。类似地,若 u^i 发生在 W_{p_i} 之后,则它一定在时间序上发生在 W_{A_i} 内的所有操作之后。

通过此定理,可以根据活动和执行窗口来导出所需的时间序。

4 TGO 图的构造及结果检测

和其他一致性模型的 TGO 图类似,在 ARM 存储模型所对应的 TGO 图中,结点所代表的访存操作执行时间信息可由结点的 bound 信息推导出来,结点之间的时间序关系边也可以从进入时间和提交时间推导出来,但是采用的有向边生成规则不同,不再有处理器边规则,而是增加了单机和同步边规则,其他生成边规则也需做相应的改动。除了文献[9]定义的生成规则外,ARM 存储模型扩充的有向边的生成规则分别是:①单机边规则,即若操作 u 和 v 在同一进程,根据 ARM 存储模型 u 在单机序上先于 v ,则加一条从 u 到 v 的边;②同步边规则,即若操作 u 和 v 在同一进程,根据 ARM 存储模型 u 在同步序上先于 v ,则加一条从 u 到 v 的边。TGO 图的构造算法也需要做如下相应的修改:

ARM 存储模型的 TGO 图构造算法

单机和同步边规则

for each processor

for each operation u in the processor

for each operation v in the pending window of u

if ($(u \xrightarrow{UP} v$ or $u \xrightarrow{SYN} v)$)

```

    and  $f(\text{type}(u), \text{type}(v)) == \text{true}$ )
    add_edge( $u, v$ )
  end if
end for
end for
end for

```

在构建好 TGO 图后,就可以在 TGO 图上检测环。这一过程是通过 TGO 图上的深度优先搜索和回溯方法完成的。

5 工具和实验结果

根据前面的算法,我们对 MOTEC 验证工具进行了扩充和改进,实现了一个检测基于 barrier 的弱存储模型的验证工具原型。改进工具分为随机指令发生模块、支持 ARM 处理器的性能计数器记录模块和结果分析模块 3 个部分。性能计数器记录模块和多核处理器系统底层相关,使用 ARM 的程序计数器来记录和推算每次扫描的边界信息。各种访存指令流由工具的随机指令发生模块随机生成,读写指令的比例和访存指令地址的条数也可由操作者根据实际情况进行配置。结果分析模块分别从 ARM 并发程序指令流执行序列和 ARM 处理器程序计数器记录的指令窗口信息接受输入,然后根据 ARM 存储模型的生成规则和指令窗口信息建立 ARM 并发程序指令执行序列的顺序关系图。最后,使用深度优先搜索算法进行遍历,判断生成的图中是否存在环:如果存在环,说明该次执行违反了 ARM 存储模型。

改进工具使用 C++ 语言实现,可以在运行时动态调整指令流的长度参数,具有很好的扩展性。利用支持 ARM MPCore 的模拟器进行了实验,并用手工方法在指令流执行序列里注入几个错误,以验证程序结果是否违反 ARM 存储模型。实验结果表明,改进工具能够正确发现上述注入错误,可以有效检测 ARM 多核处理器的存储模型。由于 ARM 模拟器的性能有限,所以我们使用该改进工具在 32 GB 内存的 IBM 服务器上运行指令达 7 000 条的操作序列,实验结果如图 3 所示,可见运行时间与指令条数成线性关系,说明改进工具的算法复杂度与指令条数成线性关系,可以快速有效地验证指令条数更多的并发程序。

6 结 语

ARM 是当前最主流的一种嵌入式处理器架构,

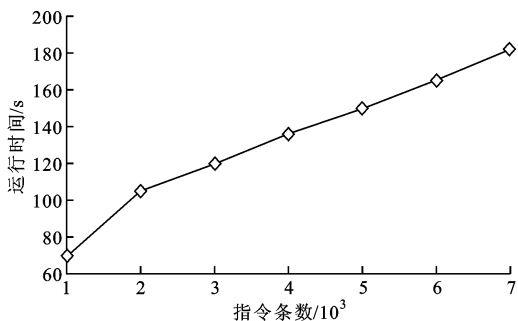


图3 改进工具的验证执行时间分析

现有的多种验证或检测存储模型的工具均不能有效地检测 ARM 存储模型。我们研究了 ARM 存储模型的快速检测方法,提出了 ARM 存储模型的形式定义和检测算法。在此基础上对 MOTEC 验证工具进行了扩充,开发了支持 ARM 处理器存储模型的验证工具。改进后的工具保持了 MOTEC 工具的低时间复杂度特点,能够快速有效地检测 ARM 存储模型。但是,这种改进工具也存在缺点,即 ARM 存储模型中的一些执行序在检测阶段无法直接观察到,可能会漏检一些错误,我们准备对此进行改进,研究更加完备的检测方法。

参考文献:

- [1] ARM. ARM architectural reference manual [M]. Cambridge, UK: ARM, 2008.
- [2] COLLIER W W. Reasoning about parallel architectures [M]. Upper Saddle River, NJ, USA: Prentice-Hall, 1992.
- [3] Compaq Computer Corp. Alpha architecture handbook [M/OL]. (1998-10-04) [2012-04-01]. <http://www.cs.cmu.edu/afs/cs/academic/class/15347-s98/public/doc/alpha-ref.pdf>.
- [4] Intel Corporation. Intel 64 architecture memory ordering white paper [R]. San Francisco, CA, USA: Intel Corporation, 2007.
- [5] HANGAL S, VAHIA D, MANOVIT C, et al. TSO-tool: a program for verifying memory systems using the memory consistency model [J]. SIGARCH Computer Architecture News, 2004, 32(2): 114-123.
- [6] MANOVIT C, HANGAL S. Efficient algorithms for verifying memory consistency [C]//Proceedings of the 17th Annual ACM Symposium on Parallelism in Algorithms and Architectures. New York, USA: ACM Press, 2005: 245-252.
- [7] ROY A, ZEISSET S, FLECKENSTEIN C, et al. Fast and generalized polynomial time memory consistency

- verification [C] // Proceedings of the International Conference on Computer Aided Verification. Berlin, Germany: Springer, 2006: 503-516.
- [8] MANOVIT C, HANGAL S. Completely verifying memory consistency of test program executions [C] // Proceedings of the 12th International Symposium on High-Performance Computer Architecture. Washington DC, USA: IEEE Computer Society Press, 2006: 166-175.
- [9] CHEN Yunji, LV Yi, HU Weiwu, et al. Fast complete memory consistency verification [C] // Proceedings of the 15th International Symposium on High-Performance Computer Architecture. Washington DC, USA: IEEE Computer Society Press, 2009: 381-392.
- [10] 王朋宇, 陈云霄, 沈海华, 等. 片上多核处理器存储一致性验证 [J]. 软件学报, 2010, 21(4): 863-874.
- WANG Pengyu, CHEN Yunji, SHEN Haihua, et al. Memory consistency verification of chip multi-processor [J]. Journal of Software, 2010, 21(4): 863-874.
- [11] LV Zheng, CHEN Hao, CHEN Feng, et al. Fast verification of memory consistency for chip multi-processor [C] // Proceedings of the Seventh International Conference on Computational Intelligence and Security. Washington DC, USA: IEEE Computer Society, 2011: 1497-1502.
- [12] 吕正, 陈昊, 陈峰, 等. MOTEC: 一个存储一致性模型验证工具 [J]. 计算机工程, 2012, 38(11): 242-246.
- LV Zheng, CHEN Hao, CHEN Feng, et al. MOTEC: a tool for quick verification of memory consistency model [J]. Computer Engineering, 2012, 38(11): 242-246.

(编辑 葛赵青)

(上接第54页)

参考文献:

- [1] VAPNIK V N. The nature of statistical learning theory [M]. New York, USA: Springer, 1995.
- [2] PÉREZ-CRUZ F, CAMPS-VALLS G, SORIA-OLIVAS E, et al. Multi-dimensional function approximation and regression estimation [M] // Lecture Notes in Computer Science: Vol. 2415. Berlin, Germany: Springer-Verlag, 2002: 757-762.
- [3] SÁNCHEZ-FERNÁNDEZ M, DE-PRADO-CUMPLIDO M, ARENAS-GARCÍA J, et al. SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems [J]. IEEE Transactions on Signal Processing, 2004, 52(8): 2298-2307.
- [4] KANG Z, GRAUMAN K, SHA F. Learning with whom to share in multi-task feature learning [C] // Proceedings of the 28th International Conference on Machine Learning (ICML). New York, USA: ACM, 2011: 521-528.
- [5] 胡运权, 郭耀煌. 运筹学教程 [M]. 3版. 北京: 清华大学出版社, 2007.
- [6] CAWLEY G C, TALBOT N L C. Preventing overfitting in model selection via Bayesian regularisation of the hyper-parameters [J]. Journal of Machine Learning Research, 2007, 8: 841-861.

- [7] 毛文涛. 支持向量回归机模型选择研究及在综合力学环境预示中的应用 [D]. 西安: 西安交通大学, 2011.

[本刊相关文献链接]

- 司刚全, 娄勇, 张寅松. 一种鲁棒最小二乘支持向量机及其在软测量中的应用. 2012, 46(8): 15-21.
- 梁炎明, 刘丁. 规则递归 T-S 模糊模型及其辨识方法. 2012, 46(8): 54-58.
- 吴德, 刘三阳. 支持向量域多分类器. 2012, 46(6): 87-91.
- 丁要军, 蔡皖东. 采用两阶段策略模型 (KTSVM) 的 P2P 流量识别方法. 2012, 46(2): 45-50.
- 杨宏晖, 戴健, 孙进才, 等. 用于水声目标识别的自适应免疫特征选择算法. 2011, 45(12): 28-32.
- 贾立新, 郑伊飞, 曹晖. 利用极值蚁群优化的制粉出力建模变量选择算法. 2011, 45(10): 7-12.
- 丁军平, 蔡皖东. 面向元信息分类的支持向量机改进技术. 2011, 45(8): 37-41.
- 丁晓剑, 赵银亮. 优化极限学习机的序列最小优化方法. 2011, 45(6): 7-12.
- 李志雄, 严新平. 独立分量分析和流形学习在 VSC-HVDC 系统故障诊断中的应用. 2011, 45(2): 44-48.
- 张周锁, 侯照文, 孙闯, 等. 应用粒计算的混合智能故障诊断技术研究. 2011, 45(1): 48-53.

(编辑 荆树蓉 武红江)