

基于布隆过滤器的事务存储架构中的高速缓存

时 磊^{1,2}, 杨 骅², 王红梅^{1,2}, 张铁军², 王东辉²

(1 中国科学院 研究生院, 北京 100039; 2 中国科学院 声学研究所, 北京 100190)

摘 要: 在基于硬件的事务存储多核处理器中, 高速缓存具有暂存事务执行结果、检测事务间冲突以及当发生冲突时解决冲突的功能, 是系统的核心模块. 为了简化上述功能, 研究并设计了一种基于布隆过滤器的高效缓存结构, 提升了事务的执行效率, 并且新增的硬件开销也比较小.

关键词: 多核处理器; 事务存储; 布隆过滤器; 高速缓存

中图分类号: TP336

文献标识码: A

文章编号: 1000-7180(2011)03-0141-03

A High-Performance Cache Based on Bloom Filter for Transactional Memory

SHI Lei^{1,2}, YANG Hua², WANG Hong-mei^{1,2},
ZHANG Tie-jun², WANG Dong-hui²

(1 Graduate School of Chinese Academy of Sciences, Beijing 100039, China;

2 Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: In multiprocessors based on hardware transactional memory, cache is the crucial functional unit and it is used to address following functions: storing speculative results of transactions, detecting conflicts between transactions and resolving the conflicts. To simplify these functions, a high-performance cache based on bloom filter is proposed in this paper, which improves the efficiency of transactions with lower hardware cost.

Key words: multiprocessor; transactional memory; Bloom filter; cache

1 引言

随着集成电路技术的不断进步, 多核处理器技术得到了迅速的发展. 在共享存储的多核处理器中, 每个处理器内核看到的都是一个统一的地址空间, 操作原子性对于保证并行程序的正确执行至关重要. 多核处理器通常采用互斥的锁(lock)结构来实现并行程序的同步和通信, 进而保证不同处理器操作的原子性. 但是基于锁的同步机制可扩展性不好, 很难正确且有效的使用, 而且容易产生死锁、活锁、优先级倒置等现象^[1].

近年来提出的事务存储多核架构可以保证操作的原子性, 同时避免锁结构产生的问题. 所谓事务, 是指由一个线程执行的包括若干访存操作的指令序

列. 事务具有原子性和串行性的特点^[2], 采用事务存储模型的多核存储器, 可以降低并行编程的复杂度, 并且事务存储系统中自动的冲突检测机制, 能够保证程序执行的正确性. 事务一致性(Transaction Memory Coherence and Consistency, TCC)^[3]是一种高效事务存储架构. 它将程序拆分为事务, 通过在事务提交的时候广播事务执行的结果, 降低了和存储一致性以及缓存一致性相关的硬件复杂度. 在 TCC 中, 缓存具有暂存事务执行的结果、检测事务间冲突以及当发生冲突的时候解决冲突的功能, 是 TCC 系统的核心模块. 但是 TCC 中采用的缓存结构导致检测事务冲突的机制复杂, 而且事务的大小受缓存容量的制约, 同时事务提交和事务撤销时的操作复杂, 事务执行效率低.

文中设计了一种基于布隆过滤器的新型缓存结构 BTC(Bloom Transactional Cache),它对 TCC 中的高速缓存结构进行了改进,简化了检测和解决事务冲突的机制,提高了事务的执行效率,同时新增的硬件开销也很小.

2 BTC 结构设计

BTC 的结构如图 1 所示,它主要包含两个部分:用于暂存事务中间结果的缓存,以及用于压缩地址、检测及解决事务冲突的过滤器 TBFILTER(Transactional Bloom Filter).

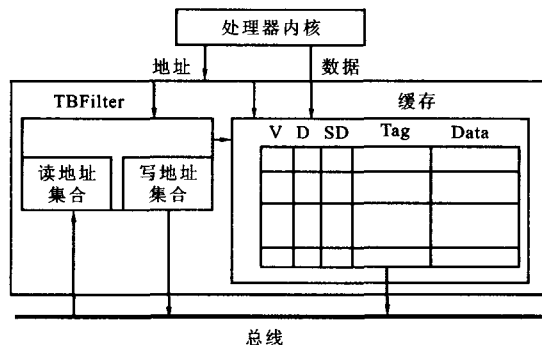


图1 BTC 的结构图

BTC 中的缓存除了基本的有效位 V(valid)、修改位 D(dirty)、tag 段和数据段外,对于缓存中的每一行,添加了事务写状态位 SD(speculative dirty),该状态位置位表示该行中的某个数据块在事务执行的过程中被推断的修改.

TBFilter 的设计基于布隆过滤器^[4-5],它的功能是用来识别当前事务的写地址集合是否和其他事务的读地址集合产生冲突.布隆过滤器是一种优秀的基于哈希映射的数据结构,它可以简洁地表示集合中的元素并支持对某个元素是否属于当前集合的快速查询.布隆过滤器可能存在错误定位的情况,但当它用于检测事务冲突的时候,只会影响程序执行的性能,不会影响程序的正确性,而且可以证明,这种错误定位的几率很小^[4].

3 BTC 工作原理

BTC 中事务执行和提交的流程如图 2 所示.

3.1 事务中间结果暂存

事务由硬件推断执行,推断执行产生的中间结果暂存在 BTC 中:当事务执行的过程中执行 store 指令时,如果该指令在缓存中命中,则该缓存的 SD 位置 1,表示它被事务推断修改;如果该指令在缓存中没有命中,则从存储器中或者其他处理器的缓存

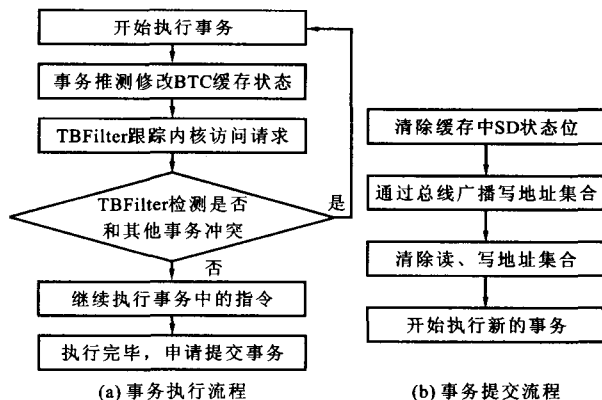


图2 BTC 中事务执行和提交流程

中读入相应的数据,进行修改并置位状态位.

如果事务顺利提交,则缓存中被事务修改的数据块对应的 SD 状态位被清零,此时这些数据可以响应来自其他处理器的读请求.而在 TCC 提出的缓存结构中^[3],当事务执行完毕的时候,必须将事务修改过的数据全部写入到存储器中,这样就会大量的占用总线的带宽,同时不必要的写入操作,也会增加系统的功耗.

3.2 事务冲突检测

在事务执行的过程中, TBFilter 以缓存中的数据行为单位,跟踪这些数据行的地址,并将他们压缩成写地址集合 W 和读集合 R. TBFilter 在 Bulk^[5]的基础上进行了改进,压缩方法如图 3 所示.具体的压缩过程是:每个传递到 TBFilter 的地址在进行地址序列变换之后被分成若干段,对变换产生的中间结果进行除留余数的 Hash 映射,将映射结果添加到读、写地址集合中.

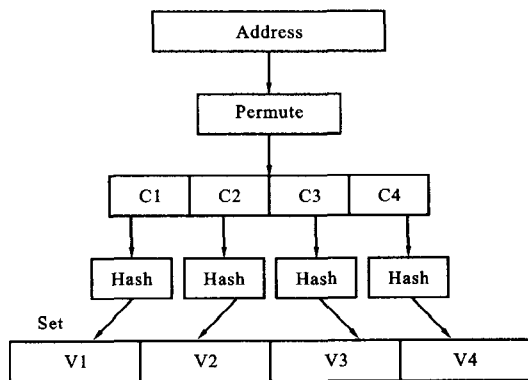


图3 TBFilter 的工作机制

当事务 A 执行完毕需要提交的时候,就将其产生的写地址集合 Wa 传递到其他事务,例如事务 B 的 BTC 处,和 B 的读地址集合 Rb 进行比较,如果 Wa 和 Rb 中有相同的元素,则表示事务间的执行顺序违反了 RAW 的原则,事务 B 需要被撤销.

采用压缩的方式保存事务读写的地址:(1)简化了事务提交的方式,在事务提交的时候,只用将相应的读地址集合和写地址集合清零,并将 SD 位清零即可;(2)扩大了事务中可以执行的指令数目,通过压缩地址的方式,缓存中就只用保存事务写操作产生的中间结果;而对于读操作的信息,则可以通过压缩产生的读地址集合保存,这样就扩大了处理器核可以执行的事务程序的大小;(3)简化了冲突检测和提交的机制,在事务提交的时候,只用将该事务产生的写地址的压缩集合通过总线传递给其他处理器进行比较,同时将缓存中的 SD 位统一清零,之后就可以立刻处理下一个事务并检测新的事务冲突。

3.3 事务冲突处理

当事务 A 提交的时候,如果发现和正在执行的事务 B 相冲突,则必须撤销 B 产生的中间结果,事务 B 从起点重新开始执行. 由于在缓存中设置了 SD 状态位,所以相对于 Bulk 方法^[5],撤销事务 B 中间结果的操作十分简单,只用将所有 SD 位置位的缓存行的 V 和 SD 状态位清零. 而在 Bulk 中,需要增加复杂的地址扩展逻辑.

4 实现结果及比较

4.1 参数设置

在具体实现 BTC 时,选用的缓存参数为:缓存的容量为 64KB,每个缓存行包含 16 个字节,四路组相连,读写时间分别为两个周期,采用写回策略,每个缓存行除了包含有效位 V、修改位 D 外,还添加了状态位 SD. TBFILTER 输出宽度为 512bit 的读、写地址变量.

4.2 结果分析

首先基于 SPEC CPU2000 的测试程序,利用 MATLAB 对 TBFILTER 和 Bulk 中的地址压缩方式进行比较,比较的内容是当对包含多个访存地址的集合进行压缩时,这两种方法产生的地址压缩向量中置 1 的位数,统计结果如表 1 所示. 可以看出, TBFILTER 和 Bulk 都维护了被测数据集合的信息量.

表 1 两种方法地址压缩结果的对比

	不同地址数	Bulk 置 1 的位数	TBFILTER 置 1 的位数
applu	44	44	44
bzip	40	40	40
sixtrack	39	39	39
gzip	16	16	16

接着对 TBFILTER 和 Bulk 压缩结果的分布情况进行了比较. 图 4 显示的是 applu 测试程序产生的

地址压缩集合的分布示意图. 横坐标表示压缩生成地址向量,纵坐标表示对地址向量中的每一位被置位的次数. 可以看出,文中方法产生的映射数据集合更加均匀,这样有利于降低采用布隆过滤器进行集合元素识别时的误判率^[4].

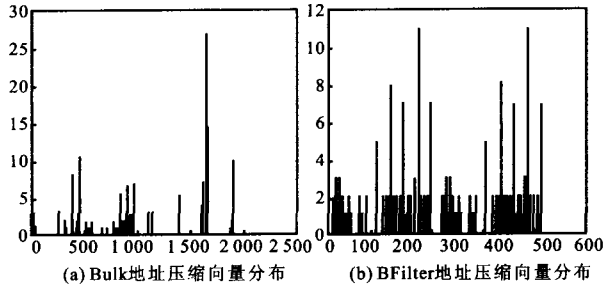


图 4 地址向量压缩分布对比图

采用 SMIC 0.18 微米的工艺对于缓存、Bulk 和 TBFILTER 进行综合,综合的结果如表 2 所示. 其中所示的面积和功耗比例为 Bulk 或者 TBFILTER 占整个缓存的比重. 可以看出, TBFILTER 的面积仅为 Bulk 的 29.48%,功耗仅为 Bulk 的 29.34%.

表 2 两种方法的具体实现对比

	面积/ μm^2	比例/%	功耗/mW	比例/%
Bulk	2 303 802	26.1	76.64	39.1
TBFILTER	679 202	9.4	22.49	15.6

5 结束语

具有保存事务中间结果、检查事务冲突的高速缓存是事务存储的多核处理器结构的核心部件. 文中提出并实现了一种基于布隆过滤器的高速缓存结构. 该方法和传统的方法相比,简化了冲突检测的机制、提升了事务的执行效率,同时新增单元的面积和功耗也都很小.

参考文献:

[1] Herlihy M, Shavit N. The art of multiprocessor programming[M]. San Fransisco: Morgan Kaufmann, 2008.

[2] Larus J, Rajwar R. Transactional memory [M]. Toronto: Morgan &. Claypool, 2007.

[3] Hammond L, Wong V, Chen M, et al. Transactional memory coherence and consistency [C]// Proceedings of the 31st Annual International Symposium on Computer Architecture (ISCA'04). Germany: Munich, 2004.

[4] Bloom B. Space/time trade-offs in hash coding with allowable errors [C]// Communications of the ACM. Switzerland: ACM, 1970.

数据维数.

表 2 主成分分析结果

	Eigenvalue	Difference	Proportion	Cumulative
1	210.747283	116.373990	0.4779	0.4779
2	94.373293	38.826544	0.2140	0.6919
3	55.546749	41.253150	0.1260	0.8178
4	14.293599	9.990312	0.0324	0.8503
	...			
27	1.478550	0.083004	0.0034	0.9770
	...			

使用主成分分析所得的 27 个点作为建立指令模板的有效点,重新指令模板分析实验,实验结果如表 3 所示.

表 3 主成分分析有效点选取匹配实验结果

指令	识别									正确率
	I ₁	I ₂	I ₃	I ₄	...	I ₃₅	I ₃₆	I ₃₇		
I ₁	63	1	1	0	...	4	2	1	63%	
I ₂	1	68	1	2	...	3	0	2	68%	
I ₃	1	0	78	1	...	3	0	0	78%	
I ₄	1	1	1	78	...	1	1	1	78%	
...	...	...	...	...	...	...	...	...	...	
I ₃₅	0	2	1	1	...	89	1	0	89%	
I ₃₆	1	0	0	0	...	5	62	2	62%	
I ₃₇	1	1	1	0	...	0	1	66	66%	

通过实验对比可以看出,使用主成分分析进行有效点选取可以大大提高指令模板识别率.

4 结束语

结合逆向工程思想和模板攻击原理,利用芯片功耗泄漏信号为芯片指令以及程序执行路径构建模板,将目标芯片上获取的旁路信号与模板进行匹配以推断芯片中执行的指令代码.通过对 AT89C52

单片机上运行单周期指令的指令模板分析试验,证明了旁路模板指令分析的可行性与有效性.并通过主成分分析提取有效点,极大提高模板匹配的效率.

参考文献:

[1] Paul Kocher. Timing attacks on implementations of dif-fie-hellman, RSA, DSS, and other systems[C]// Neal Koblitz. Advances in Cryptology, Proceedings Crypto '96, LCNS. New York: ACM, 1996:104-113.

[2] 张鹏,邓高明,邹程,等. 差分功率分析攻击中的信号处理与分析[J]. 微电子学与计算机,2009, 26(11): 1-4.

[3] Chari S, Rao J R, Rohatgi P. Template attacks[C]// Proceeding of the Workshop on Cryptographic Hardware and Embedded Systems (CHES'03). Cologne, Germa-ny: Computer Science, 2003:13-28.

[4] 邓高明,张鹏,赵强,等. 针对密码芯片的电磁模板分析攻击[J]. 微电子学与计算机,2010, 27(1): 1-4.

[5] Dennis Vermoen. Reverse engineering of Java Card ap-plets using power analysis [D]. Netherlands: TU Delft, 2007.

[6] Sung-Mo Kang, Yusuf Leblebici. CMOS 数字集成电路一分析与设计[M]. 王志功, 窦建华, 译. 北京:电子工业出版社,2005: 97-98.

[7] 梅长林,周家良. 实用统计方法[M]. 北京:科学出版社,2006:53-65.

作者简介:

吴恒旭 男,(1983-),硕士研究生.研究方向为信息安全.

陈开颜 女,(1970-),博士,硕士生导师.研究方向为信息安全.

邓高明 男,(1983-),博士研究生.研究方向为电磁信息检测与主动防护技术.

张 鹏 男,(1976-),博士研究生.研究方向为电磁信息检测与主动防护技术.

邹 程 男,(1982-),博士研究生.研究方向为电磁信息检测与主动防护技术.

(上接第 143 页)

[5] Ceze L, Tuck J, Cascaval C, et al. Bulk disambiguation of speculative threads in multiprocessors [C]// Proceedings of the 33st Annual International Symposium on Computer Ar-chitecture(ISCA'06). Boston: ACM, 2006.

作者简介:

时 磊 男,(1983-),博士研究生.研究方向为 VLSI 信号处理.