升级 RISC-V 的指令级仿真器 Spike 的缓存模型

李真真，宋威

(中国科学院信息工程研究所，北京市海淀区，邮编 100093)

摘　要：基于研究的需要，我们正在着手升级 RISC-V 的指令级仿真器 Spike 的缓存模型。本文将首先简单介绍 Spike 指令级仿真器，然后分析 Spike 原有缓存模型的优点和不足。我们希望利用 Spike 仿真器进行多核一致性缓存的快速研究，发现现有的缓存架构不能满足实际研究的需要，因此我们提出升级 Spike 的缓存模型。本文的后续部分介绍了新版缓存模型的功能特点以及我们的升级进展。

关键词： Spike；指令级仿真器；自定义缓存模型；缓存一致性

1　Spike 介绍

Spike[1] 是 RISC-V 的指令级仿真器（ISS，Instruction set simulator）。Spike 是 RISC-V 基金会[3]所指定的 RISC-V ISA 标准实现（Golden model）。所有 RISC-V ISA 的扩展和更新，事先都会在 Spike 上实现、验证并评估。所以，Spike 是支持 RISC-V ISA 最完整的 RISC-V ISA 运行环境。

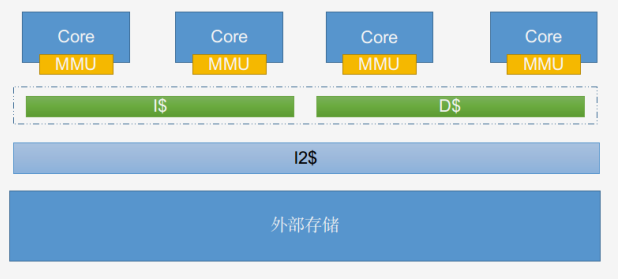
从功能上来说，Spike (riscv-isa-sim) 是一个基于 C/C++开发的指令级仿真器，为 RISC-V 体系架构提供了开发和测试的执行环境。Spike 通过软件实现了一个通用的指令执行环境，包括内存系统、虚拟和物理地址空间、支持 RISC-V 所有 ISA 扩展的处理器流水线和最基本的输入输出接口与磁盘系统（配合使用 riscv-pk[2]）。同时，Spike 支持很多种调试机制，比如说使用 GDB 调试器调试 Spike 本身。另外，配合使用 OpenOCD 之后可以和 GDB 直接调试运行于 Spike 之上的 RISC-V 执行程序。这些都是 FPGA 等真实 RISC-V 处理器很难做到的。目前 Spike 支持的 RISC-V 基础指令集与扩展指令集如下：

基础指令集：RV32I 和 RV64I。

扩展指令集：A (原子操作指令)、M (定点乘除法指令)、C (压缩指令)、F (单精度浮点指令)、D (双精度浮点指令)、 Q (四倍精度浮点指令)和 V (向量计算指令)。

2　Spike 中的缓存模型

在非 RISC-V 的计算机平台上（比如 x86-64），开发者无法直接运行 RISC-V 的可执行程序。Spike 通过将 RISC-V 可执行程序（一般为 ELF 格式）加载到主机内存系统中，然后在主机系统上模拟一个 RISC-V 的执行环境，以动态二进制翻译的方式动态执行 RISC-V 程序。



Spike 支持模拟两级的缓存结构：用于缓存指令的一级指令缓存（L1 I$），用于缓存数据的一级数据缓存（L1 D$），和统一的二级缓存（L2$）。Spike 支持多核，每一个核有一个负责虚拟地址、物理地址和主机地址之间地址映射的内存管理单元（MMU）。所有的核连接到 L1 层的 I$和 D$的对象。但需要注意的是，Spike 的这种缓存连接方式，或者说仿真方式，和实际多核处理器中的一级缓存结构完全不同。在仿真多核系统时，多核共享同一组的一级缓存，而不是各有其私有的一级缓存，这就导致直接使用 Spike 的缓存模型在研究多核系统的缓存特征时，存在和实际硬件系统的不可忽略的偏差。

此外，Spike 的 MMU 也没有按照实际硬件系统访问缓存。每个核的私有内存管理单元（MMU）会把虚拟地址（VA）翻译成物理地址（PA）和其对应的主机内存地址（HA），同时将 PA 和 HA 对应 VA 的偏移量存入一个软件 TLB 模块。在取指令和读写数据时，MMU 会访问 TLB 直接获得指令的主机地址，从主机内存直接读取，仅将内存访问请求发给缓存模型而并不如真实硬件一样，在缓存模型中存储数据。该 TLB 也并没有在仿真实际处理器中的 TLB 模块。当 TLB 发生项缺失时，对 TLB 的回填并不会访问一级缓存，而是直接从主机内存总读取页表。也就是说，缓存模型并没有记录任何页表的访问行为，从而进一步扩大了和真实硬件系统的区别。

Spike 缓存模型和实际硬件系统的差别其实是为了提高仿真器的仿真速度。为了能快速仿真指令执行，Spike 只对真实的硬件功能进行了部分的模拟，而对很多不影响指令执行的部分进行了简化处理。我们通过分析后发现，Spike 中的缓存模型未实现以下的功能：

1.现有的 Cache 模型比较简单。缓存所使用的缓存结构为最经典的多路组相联结构，用户不能选择其他的缓存结构、修改置换算法、缓存映射算法等等。

2.没有私有一级缓存的概念，且无法支持多核的缓存一致性。

3.页表不存储在缓存中。

4.无真实的 TLB。Spike 中的 TLB，只有为了加快仿真速度的软件 Cache，存储的是主机地址相对于虚拟地址偏移量。该软件 TLB 大大加快了运行的速度，但未模仿真实硬件 TLB 行为。

5.性能指标数据较少，目前的 Spike 缓存的指标包括未命中率（Miss rate）、读写的访问次数、读写的访问时间。而 Cache 的相关研究中，需要提取更多维度的性能指标。

3 Spike 中的缓冲模型升级的研究工作

因为目前 Spike 中的缓存模型存在上述问题，同时为了在 Spike 上进行 Cache 的基于多核一致性的缓存测信道攻击的研究，我们新建了一个支持多核一致性的缓存模型，并开始了对 Spike 仿真器中缓存模型的升级。新的缓存模型支持以下的特性：

1.可以根据需要自定义 Cache 的结构。

通过编译时的参数配置（或者 Spike 参数列表），用户可以自定义缓存的结构特征。除了缓存结构为最经典的多路组相联结构，新的缓存模型还支持 skewed cache[4]等。用户也可以通过可扩展的缓存框架编程实现其他的缓存结构。同时缓存的层数、大小、路数、块大小等参数也支持配置。另外，可扩展的缓存框架为将来支持智能预取单元、置换缓存（victim cache）和流缓存（stream cache）提供了可能性。

2.目前二级缓存模型中已加入了随机缓存组映射算法。Spike 通过最简单的位选取（bit selection）算法选择缓存组，即直接截取地址的位作为缓存的路组的索引。最近的安全研究发现，随机映射算法[5]相较于位选取算法，大大提升了处理器对抗缓存测信道攻击的能力。新的缓存模型已支持该随机映射算法。

3.目前新的缓存模型支持二级缓存和三级缓存。支持目前较为流行的三级缓存的设置：L1/L2 为每个 CPU 核独享，而 L3 或者 LLC（last level cache）则由所有核共享。

4.加入任意替换算法。已实现的替换算法包括先进先出算法（FIFO）、LRU 算法（Least Recently Used）和随机替换算法。用户可编程扩展自己的替换算法。

5.支持多核一致性（Cache-coherence）。在多核的缓存模型中，由于内存中的数据有且只有一份，但多个核很可能在此基础上创建属于自己的缓存。对于内存的读访问来说，多份数据并不会影响到一致性，但一旦某个核修改了数据，一致性问题则会凸显。新的缓存模型目前支持多级缓存之间基于广播的 MSI 缓存一致性协议。计划将陆续实现基于目录的缓存一致性协议。

6.支持缓存的多维度特征提取和本地保存，便于后续进行数据分析。目前 Spike 的缓存统计只有 3 个简单指标且只能在交互式命令中使用，对于大规模的实验，无法满足要求。目前缓存模型已经有的指标包括：每一个缓存的总访问次数、总置换次数、访问率、命中率、置换率；每一个缓存组的总访问次数、总置换次数、访问率、命中率、置换率。将来要实现（即可选）的：某一个地址在缓存中的命中率、置换率，某一个缓存或者缓存组的地址分布等等。同时这些文件被格式化的（基于 json）存放在本地，方便对实验数据进行整理和事后分析。

目前模型开发的工作基本上完成，同时已经在组内用于缓存研究。就在今年，该模型已经被成功用于分析缓存测信道攻击在缓存中的统计特性，进而优化缓存测信道攻击中的缓存置换集（eviction set）的自动生成算法，相关结果已经发表在 2019 年的 RAID 会议上[6]。现阶段的工作集中在替换 Spike 的缓存模型，完成仿真系统的功能联调和测试。未来我们也将支持模拟硬件行为的 TLB，因为 TLB 和页表对缓存造成的存储压力在分析不同负载模型的缓存性能以及缓存测信道攻击特征的研究中具有重要的意义。

4 参考文献

[1]RISC-V.Spike 的主代码库[EB/OL].[2019/04/01] https://github.com/riscv/riscv-isa-sim

[2] RISC-V.riscv-pk 的主代码库 [EB/OL].[2018/02/15] https://github.com/riscv/riscv-pk

[3] RISC-V 基金会.RISC-V 的官网[EB/OL].https://riscv.org

[4] Moinuddin K. Qureshi. New attacks and defense for encrypted address cache. In Proceedings of the International Symposium on Computer Architecture, June 22–26, 2019.

[5] Moinuddin K. Qureshi. CEASER: Mitigating conflict based cache attacks via encrypted-address and remapping. In Proceedings of the Annual IEEE/ACM International Symposium on Microarchitecture, pp. 775–787, IEEE, 2018.

[6]Wei Song and Peng Liu. Dynamically finding minimal eviction sets can be quicker than you think for side-channel attacks against the LLC. In Proceedings of the International Symposium on Research in Attacks, Intrusions and Defenses, September 2, 2019.