

Summary

许多现代多核计算机系统都支持硬件共享内存。在共享内存系统中，每个处理器内核可以读取和写入单个共享地址空间。高速缓存一致性协议管理着各种高速缓存中数据的读写权限，是保证多核系统共享高速缓存正确运行的重要组成部分。

高速缓存一致性协议的设计非常具有挑战性，特别是由于大量的竞争和边缘情况。验证它们的正确性是这个过程中必要但非常困难的一步。目前，非常复杂的协议已经被开发出来并验证，但是这仍然是一个活跃的研究领域。在本次实验中，我们将设计和验证一个多核系统的缓存一致性协议。

为了验证我们的一致性协议，我们将使用 Murphi，一个正式的验证工具。Murphi 采用模型检查来验证指定的一致性协议的正确性。模型检查是一种自动化技术，在给定系统的有限状态模型和一组形式属性的情况下，在该模型中检查属性是否适用于系统的所有有效状态。更具体地说，我们将描述与 Murphi 描述语言中的相关协议对应的有限状态机，并列举一组期望的属性或不变量。Murphi 验证器将系统地枚举整个状态空间，并检查指定的不变量是否被违反。

我们已经在资源列表中提供了必要的材料，你可以下载 CMurphi 5.4 并进行安装，这是 Murphi 的一个扩展版。同之前的实验一样，这个实验需要你单独完成，你可以通过阅读附带的 Murphi 指导内容来学习使用，或与同学讨论你遇到的问题。但是请注意按时提交你的成果，最终程序将会通过查重系统的检测。

Setting Up

请首先从实验平台下载本次实验需要的验证工具 CMurphi 5.4。你可以将它安装至任何位置，因为本次实验你将不需要使用 Pin。从实验平台下载实验模板 lab3.m，

安装方法：进入 CMurphi 目录下的 src 文件夹，运行 make 指令进行编译

验证你的安装：进入 Cmurphi 目录下 ex/toy，将 pingpong.m 文件复制到 src 下。这是

一个最简单的 Murphi 示例程序，你可以尝试阅读这个示例来获得一些灵感。之后进入 src 目录，运行下列指令：

- `./mu pingpong.m`
- `g++ -ggdb -o pingpong.o pingpong.cpp -I ../include -lm`
- `./pingpong.o`

mu 是 Murphi 编译器，它将 .m 文件转化为 .cpp 文件，然后由 g++ 编译执行。

特别说明的是，*.m 文件是使用 Murphi 描述语言编写的，其中描述了你设计的协议的有限状态机。因此，你需要将你的协议转化为有限状态机的形式，具体语法与使用方式请自行查找相关资料。

Lab Task

你将设计并验证基于失效的缓存一致性协议。您开发的协议将具有许多特征：

1. 每个处理器都有一个不与其他任何处理器共享的专用缓存。所有缓存必须通过缓存一致性协议保持一致。处理器只能发出加载和存储操作。因为这个任务只涉及缓存一致性问题，而不涉及一致性问题，所以您只关心一个存储位置（地址）。但是，您需要建立缓存冲突的模型。为此，除了加载和存储之外，还需要建模第三个操作：缓存回写。如果旧线路脏了，回写通常是由缓存冲突引起的。在任何一对加载/存储操作之间的任何时候都可能发生回写操作。如果缓存处于干净状态，则可以简单地将其设置为无效或根据您的一致性协议采取适当的操作。高速缓存替换脏行必须将行写回内存。

2. 你应该假定一致性单位等于一个单词，所有的加载和存储都读或写整个单词。

3. 除了具有高速缓存的处理器之外，系统中还有一个内存单元。内存单元有一个基于目录的缓存一致性控制器，它确保一次只有一个处理器可以写入内存块（独占所有权型协议）。目录表示对于这个任务是不重要的。你应该假设你有一个完整的目录（位向量），可以跟踪所有的共享者。

- 4...

Submit

你的最终成果将是一个三跳 MSI 协议，通过 Murphi 语言进行验证。你最好能有一份你的完整状态机的状态转换图在实验报告中。同时，Murphi 的输出应该仅有探测到的状态数量和运行时间，而不应该存在错误。

请于实验系统要求的时限之前提交本次实验要求的 lab3.m，并在报告提交的位置上传你的实验报告。

Addition

<https://arabelatso.github.io/categories/%E5%BD%A2%E5%BC%8F%E5%8C%96%E6%96%B9%E6%B3%>

Advice

我们的代码或基础架构中可能存在错误。如果你发现任何“有趣的”或“意外的”行为，这可能是我们提供的代码或基础架构中的问题。请立即将这些错误报告给技术援助。