标题：xxxxxxxx

Xx, xx, xx

（南昌大学信息工程学院，江西 南昌 330031）

摘 要：RISC-V~~(读作risk five)~~是一个开源的模块化指令架构，在基于RISC-V指令架构的低功耗移动端处理器上部署人工神经网络已经成为确实的需求与挑战。使用虚拟化技术可以实现异构模型的快速部署，但传统的虚拟化方式会对系统性能产生性能下降、资源利用率低、额外开销大等等问题。为了在资源有限的RISC-V指令架构上实现轻量级虚拟化，通过命名空间隔离进程，将异构二进制代码拦截、转换，实现异构模型快速执行。初步的实验结果表明，整个系统可以在RISC-V架构下快速部署通过深度学习框架如（如TensorFlow、PyTorch等）交叉编译的神经网络模型，并且对性能的影响比硬件虚拟化方式更小。

关键词：虚拟化，神经网络，RISC-V

~~中图分类号：TP301 文献标志码：A~~

Title：

**~~Abstract~~**~~: Placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder placeholder. placeholder~~

**~~Key~~****~~words~~**~~: placeholder placeholder placeholder~~

1. 引言

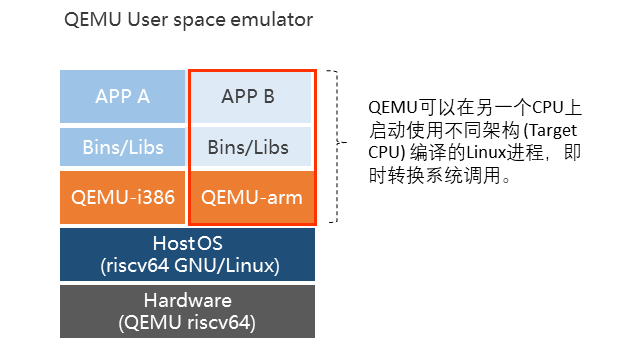
随着登纳德缩放定律和摩尔定律的终结，标准微处理器性能提升的减速已成为了既定事实，体系结构在新的黄金时代需要寻求新的前进方向。受开源软件的启发，加州大学伯克利分校提出了RISC-V（读作 risk five），即第五代RISC架构。RISC-V是一个开源的模块化指令集架构，由基本的整型指令集模块和基于整型指令集的可选拓展模块组成。模块化的设计使得研究人员可以根据特定应用定制化处理器设计。RISC-V在在工业界和学术界的应用日益广泛，良好的开源生态环境，使得研究人员可以通过众多开源工具链更轻松地构建基于RISC-V指令集架构的系统。

于此同时，深度学习在计算机视觉、语音识别、自然语言处理等领域有了越来越密集的应用，研究人员尝试在各种计算平台上部署神经网络模型，包括云端、移动设备、物联网（Internet of Thing, IoT）设备等。在服务器和桌面设备中主要采用 x86 指令架构的处理器，而移动设备和 IoT设备的处理器市场基本使用 Arm 指令架构。在各种指令架构的边缘设备上快速部署深度神经网络模型已经成为迫切的需求。

目前主流的深度学习框架（如TensorFlow、PyTorch等）并不能很好的支持RISC-V指令架构的处理器，研究人员往往需要通过构建复杂的交叉编译工具链，或者自定义指令集拓展的方式来部署神经网络模型。在Arm指令架构的边缘设备上，为了满足快速部署模型的要求，研究人员往往会使用虚拟化技术来快速构建异构模型。传统的硬件级虚拟技术模拟全部真实硬件设备，会占用宿主机大量系统资源，造成宿主机性能下降。为了节省边缘设备上有限地硬件资源，用户需要使用更加轻量级的虚拟化技术，在提供高质量的虚拟环境的同时，降低对系统性能的影响。容器技术通过名字空间 Namespace 为每个容器提供特定的命名空间，对进程实现隔离，相对于传统的虚拟机，容器技术具有更少的系统占用，更快的启动速度和更高的资源利用率。Docker是目前最为常用的容器技术，但Docker尚不支持RISC-V架构。

本文在RISC-V指令架构的硬件平台上利用QEMU实现了轻量级的虚拟化部署，结合容器化技术的思想，通过命名空间隔离系统进程，在虚拟内核中直接运行异构编译的深度神经网络模型，最终完成的工作如下：

（1）在RISC-V指令架构的硬件平台上实现轻量级虚拟化，用户可以RISC-V指令架构上快速部署异构文件；

（2）将在通用x86-64指令架构下构建的深度神经网络模型部署在RISC-V指令架构的硬件平台上，与硬件级虚拟化方式进行对比，初步验证系统的可行性与高效性。

本文的结构如下：第2节讨论了与本文相关的工作；第3节介绍了轻量级虚拟化系统整体设计；第4节展示了初步的实验结果；最后一节总结全文。

2 相关工作

Yangyang Kong 提出了AIRV，在RISC-V上使用深度学习推理的计算框架

Marcia Sahaya Louis等人自定义了RISC-V指令集拓展，并在此基础上增加了一层软件结构，执行并优化TensorFlow-Lite神经网络模型。[2]

其他的虚拟化技术（kvm/xen/docker）

图1

3 RISC-V平台上的轻量级虚拟化实现

虚拟化是将硬件资源抽象为虚拟逻辑对象的技术，包括计算机的硬件资源、存储设备和网络资源的虚拟等。虚拟化技术通过在软件和硬件层面引进虚拟化层，将软件和硬件分离。真实的硬件环境称之为宿主机，虚拟化实现的环境称之为虚拟机。在虚拟化的操作系统中执行应用程序，可以提高应用的安全性。

最早的CPU虚拟化技术是通过二进制翻译实现的，在宿主机和虚拟机之间添加一层中间层，将宿主机处理器的指令代码转换、翻译成目标处理器的指令集，捕获文件执行时所需的系统调用。VMWare 、 QEMU等均是采用的这种方法实现硬件的虚拟化。

QEMU是通过纯软件方式实现硬件的虚拟化过程，在RISC-V指令架构的硬件平台上可使用QEMU模拟外部硬件，通过这种方式可执行异构文件。

图1

我们使用 QEMU 的User-Mode模拟命名空间 Namespace，隔离Linux系统进程。在 User-Mode 下，QEMU不会仿真所有硬件。异构文件在执行时，代码传递至虚拟进程中，QEMU 将异构代码转换成本地指令架构代码，同时捕获系统调用，并将其转发至本地主机内核。其中的转换主要是通过QEMU中的TCG ( Tiny Code Generator)模块 实现的。

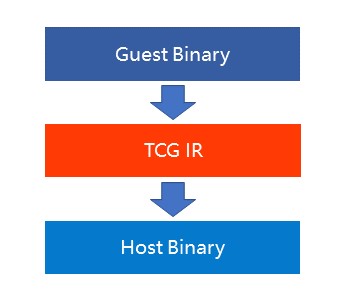
TCG 定义了一系列IR (intermediate representation)，已经翻译的代码块放在转换缓存中，并通过跳转指令将源处理器的指令集（ISA）和目标处理器的指令集（ISA）链接在一起。当`Hypervisor`（虚拟机管理程序）在执行代码时，存放于转换缓存中的链接指令可以跳转到指定的代码块，并且执行可以在不同的已翻译代码块上运行，直到需要翻译新块为止。在执行的过程中，如果遇到了需要翻译的代码块，执行动作就会暂停并回会跳回到Hypervisor，Hypervisor就会使用和协调TCG对需要进行二进制翻译的源处理器指令集（ISA）进行转换和翻译并存储到转换缓存中。

图2 TCG

4 实验设置与分析

实验平台为 QEMU 模拟的RISC-V指令架构的4核虚拟处理器，内存为2G，主频为1.7 GHz。在实验平台上通过QEMU实现轻量级虚拟化，分别运行3组不同方法的强化学习模型，并测试模型性能和运行时系统资源占用。强化学习模型模拟Cart-Pole环境，学习目标是使木棍在小车上树立的时间尽量长，采用的算法如下：

（1）随机代理 (Random Policy )

（2）交叉熵 (Cross-entropy)

（3）策略梯度 (Policy Gradient )

三组算法的复杂度依次递增，测试系统对支持复杂网络的可行性。

图3

同时我们使用QEMU实现全系统虚拟化，在实验平台上模拟一个 x86-64 位架构虚拟处理器，运行相同三组神经网络模型，测试其性能损失和系统占用。

图4

图5

5 结束语

我们使用 QEMU 在资源有限的RISC-V处理器平台上简单实现了一种轻量级的虚拟化方案，是在宿主机的进程之中模拟一个目标内核，通过 TCG 将异构模型的二进制代码转换成 RISC-V 指令集代码，同时捕获系统调用，转发至本地内核。这种做法不需要额外的硬件支持，初步的实验证明，该方法可以实现异构模型的快速部署，且比传统的虚拟化方法对性能的影响更小。

参考文献：