分类号： 密级：

UDC： 学号：

南 昌 大 学 硕 士 研 究 生

学 位 论 文

**基于RISC-V架构的容器化可重现方法研究**

**Research of Reproducible Container Based on**

**RISC-V Architecture**

崔傲

培养单位（院、系）： 数学与计算机学院

指导教师姓名、职称： 徐子晨 教授

申请学位的学科门类： 工学

学科专业名称： 计算机科学与技术

论文答辩日期：

答辩委员会主席：

评阅人：

年 月 日

书脊(提交论文电子版时，此页请删除)

|  |
| --- |
| 5cm左右  **中部崛起视野中的中东部互动发展与经济合作研究**  **姓名**  **南昌大学**  5cm左右 |

**一、学位论文独创性声明**

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南昌大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名（手写）： 签字日期： 年 月 日

**二、学位论文版权使用授权书**

本学位论文作者完全了解南昌大学有关保留、使用学位论文的规定，同意学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权南昌大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编本学位论文。同时授权北京万方数据股份有限公司和中国学术期刊（光盘版）电子杂志社将本学位论文收录到《中国学位论文全文数据库》和《中国优秀博硕士学位论文全文数据库》中全文发表，并通过网络向社会公众提供信息服务，同意按“章程”规定享受相关权益。

学位论文作者签名（手写）： 导师签名（手写）：

签字日期： 年 月 日 签字日期： 年 月 日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 论文题目 |  | | | | |
| 姓 名 |  | 学号 |  | 论文级别 | 博士□ 硕士□ |
| 院/系/所 |  | | 专业 |  | |
| E\_mail |  | | | | |
| 备注： | | | | | |

□公开 □保密（向校学位办申请获批准为“保密”， 年 月后公开）

摘 要

在进行程序调试、分布式系统构建及机器学习模型训练等应用场景中，往往需要确保程序在不同机器间反复执行能够得到可重现的结果。现有的确定性重放工作难以同时满足确定性和可移植性的要求：一是保证输出的确定性需要对程序进行的额外修改来维护系统时间或标识符的唯一性；二是可移植性差，难以跨机器执行。针对上述问题，本文对基于RISC-V架构的容器化可重现方法展开研究，主要工作如下：

第一，本文对Linux系统下的不确定性进行研究，确定不确定性来源，并就确定性重放问题的国内外研究现状展开调查与分析。

第二，基于上述问题与内容，本文在x86架构下实现一种容器化可重现方法，建立可重现容器抽象，保证容器内程序强制以可重现的方式运行，且无需对源程序进行修改，并向RISC-V平台进行移植优化。该方法通过命名空间等隔离用户进程，利用ptrace对标识符、系统调用、信号等中的非确定因素进行追踪、拦截，实现可重现的软件输出。本文还通过模拟器将可重现容器抽象移植到RISC-V架构下运行。

第三，本文使用Gem5对RISC-V硬件平台进行全系统仿真，并进行容器化可重现方法的实验验证和性能评估。评估结果表明，与xxxx相比，…………。

**关键词：**确定性；RISC-V；记录重放；容器化

ABSTRACT

……

**Key Words:**

目 录

[第1章 引言 1](#_Toc97428709)

[1.1 研究背景与意义 1](#_Toc97428710)

[1.2 国内外研究现状 1](#_Toc97428711)

[1.2.1 国外研究现状 1](#_Toc97428712)

[1.2.2 国内研究现状 1](#_Toc97428713)

[1.3 主要研究内容 1](#_Toc97428714)

[1.4 组织结构 2](#_Toc97428715)

[第2章 RISC-V指令集与可重现性研究 1](#_Toc97428716)

[2.1 RISC-V指令集 1](#_Toc97428717)

[2.1.1 模块化设计 1](#_Toc97428718)

[2.1.2 函数调用规范 1](#_Toc97428719)

[2.1.3 RV32/64特权架构 1](#_Toc97428720)

[2.2 可重现性研究 1](#_Toc97428721)

[2.2.1 确定性重放 1](#_Toc97428722)

[2.2.2 确定性模型 2](#_Toc97428723)

[2.2.3 确定性重放实现方法 2](#_Toc97428724)

[2.2.4 容器化方法中的不确定性来源 2](#_Toc97428725)

[第3章 容器化可重现方法 3](#_Toc97428726)

[3.1 基于ptrace的系统调用拦截 3](#_Toc97428727)

[3.2 可重现方法工作流程 3](#_Toc97428728)

[3.3 不确定性来源 3](#_Toc97428729)

[3.3.1 用户进程编号 3](#_Toc97428730)

[3.3.2 随机函数 4](#_Toc97428731)

[3.3.3 时间 4](#_Toc97428732)

[3.3.4 信号 4](#_Toc97428733)

[3.3.5 文件和目录 4](#_Toc97428734)

[3.4 容器化隔离方案 4](#_Toc97428735)

[第4章 基于RISC-V架构的容器化方法 5](#_Toc97428736)

[4.1 基于Gem5的RISC-V全系统仿真 5](#_Toc97428737)

[4.1.1 Gem5全系统模拟器 5](#_Toc97428738)

[4.2 RISC-V容器化方法 5](#_Toc97428739)

[4.2.1 基于命名空间的容器架构 5](#_Toc97428740)

[4.2.2 基于QEMU的模拟器 5](#_Toc97428741)

[4.2.3 动态二进制指令翻译 5](#_Toc97428742)

[4.3 RISC-V容器化方法中的不确定性来源 5](#_Toc97428743)

[第5章 实验设计与分析 6](#_Toc97428744)

[5.1 软硬件平台 6](#_Toc97428745)

[5.2实验环境配置 6](#_Toc97428746)

[5.3 基准测试集 6](#_Toc97428747)

[5.4 可重现性方法功能验证 6](#_Toc97428748)

[5.5 可重现方法性能损耗分析 6](#_Toc97428749)

[第6章 结论与展望 7](#_Toc97428750)

[致 谢 8](#_Toc97428751)

[参考文献 9](#_Toc97428752)

[附录A ××××× 10](#_Toc97428753)

[攻读学位期间的研究成果 11](#_Toc97428754)

第1章 引言

1.1 研究背景与意义

多核架构和并行程序，需要面临程序调试结果不确定的问题。

……

确定性重放的基本思想是：

……

在此基础上，增加可移植性的要求。

现有解决方案存在以下问题：

本文的工作。

1.2 国内外研究现状

1.2.1 确定性重放技术研究现状

1.2.2 RISC-V研究现状

1.3 主要研究内容

1.4 组织结构

第2章 RISC-V指令集与可重现性研究

2.1 RISC-V指令集

2.1.1 模块化设计

2.1.2 函数调用规范

2.1.3 RV32/64特权架构

除用户模式（U-mode）以外，RISC-V设计了两种具有更高权限的模式：机器模式（Machine Mode，缩写为M-mode）以及监管者模式（Supervisor Mode，缩写为S-mode）。

M-mode被设计用来拦截和处理异常。中断和异常是不确定性的主要来源。

S-mode的核心是使用基于页面的虚拟内存实现内存保护。这是一种用于更复杂RISC-V处理器上的可选模式。S-mode的权限基于U-mode和M-mode之间，不能使用M-mode下的CSR和指令。在类Unix系统，如

2.2 可重现性研究

分解为2部分：确定性及可移植性。

2.2.1 确定性重放

确定性重放也常被成为记录重放（Record and Replay），即在时间或空间上重复执行一个程序，执行的副本作为原程序的镜像，在相同的输入下应当产生与之相同的输出。数值计算型程序多次执行结果必然相同，但是涉及到系统时间、文件输入、缓冲区等因素影响的程序，执行结果会受到多种因素的影响。因此需要通过记录重放的方法使程序反复执行得出相同的结果。确定性重放具有多个应用场景：

**软件构建：**

**机器学习：**

**分布式系统：**

2.2.2 确定性模型

分析程序执行过程中的不确定性来源：1）具有不确定性的指令；2）中断与异常；3）具有随机性的系统调用等；4）线程间资源竞争；5）输入输出操作；6）线程调度等。

2.2.3 确定性重放实现方法

确定性重放可分别通过硬件支持方法和纯软件实现。

硬件方法：修改体系结构、设计专用芯片等。成本高、增加功耗、灵活性低。

纯软件方法：

1. 系统级别
2. 进程级别

2.2.4 虚拟化与容器

第3章 容器化可重现方法

在x86平台上，设计容器化可重现方法。使用纯软件用户空间的方式隔离容器中的程序，以系统调用getpid（获取进程编号PID）为例，使用ptrace监视并拦截容器中运行中程序（称为用户进程）的系统调用。

3.1 基于ptrace的系统调用拦截

定义：

ptrace()系统调用函数提供了一个进程（the “tracer”）监察和控制另一个进程（the “tracee”）的方法。并且可以检查和改变“tracee”进程的内存和寄存器里的数据。它可以用来实现断点调试和系统调用跟踪。

3.2 可重现方法工作流程

通过用户空间隔离进程，使用追踪进程tracer可以拦截被追踪用户进程tracee的系统调用，读取和写入进程B的内存与寄存器。其中可重现的元素被允许通过，不可重现的元素被进行可重现的包装或者被禁止进入容器空间。

3.3 不确定性来源

3.3.1 用户进程编号

通过命名空间隔离进程，获取惟一PID等。

3.3.2 随机函数

拦截具有随机性的系统调用函数，用简单伪随机函数替换。

3.3.3 时间

返回时间信息的系统调用，如，替换为用户进程执行的时间调用的计数。

3.3.4 信号

异步信号。

部分信号是天然可重现的：, 和。

3.3.5 文件和目录

通过隔离用户进程拥有的文件系统。

作用于正在运行的用户进程和它的子进程，改变它外显的根目录，设置后的用户进程不能够对这个指定根目录之外的文件进行访问动作，不能读取，也不能更改它的内容。

3.4 容器中的不确定性来源

……

第4章 基于RISC-V架构的容器化方法

4.1 基于Gem5的RISC-V全系统仿真

4.1.1 Gem5全系统模拟器

引导加载程序：bbl 伯克利引导程序

Linux 4.12内核版本

文件系统：BusyBox

4.2 RISC-V容器化方法

4.2.1 基于命名空间的容器架构

4.2.2 基于QEMU的模拟器

4.2.3 动态二进制指令翻译

4.3 RISC-V容器化方法中的不确定性来源

4.4 RISC-V特权级切换

第5章 实验设计与分析

5.1 软硬件平台

模拟硬件：Gen5模拟器全系统模式下，模拟多核RISC-V处理器。

软件：Linux 4.12内核版本。

5.2实验环境配置

配置流程。

5.3 可重现性方法功能验证

运行包含getid()、等的程序。

5.4 可重现方法性能损耗分析

重复调用，记录执行时间。

第6章 结论与展望

……

……

致 谢

……

xxx（学生姓名落款）

年 月

参考文献

1. 周绍森，王志国，胡德龙. 中部塌陷与中部崛起[J]. 南昌大学学报（人文社会科学版），2003，34(6): 14～18.
2. Wilfred J, Pandey V N, Nair C K K, et al. Risk sharing and industrial specialization: regional and international evidence [J]. The Economic Journal, 1998, 108(11): 1149～1161.
3. 奥桑德主编．区域整合中的中央与地方关系(上册,第1版)[M]．胡动译．南京：南京大学出版社，1981, 452～482.
4. 冯兴元．区域经济学[M]．天津：天津大学出版社，1979.
5. 惠梦君,吴德海,柳葆凯,等. 奥氏体—贝氏体球铁的发展[C]. 全国铸造学会奥氏体—贝氏体球铁专业学术会议．武汉,1986.
6. 金波. 采用并联型液压系统的水轮机调速器控制系统研究[D]. 浙江大学，1998.
7. 张楠. 一种氧化锆材料的制备方法[P]. 中国,专利文献种类, 881089073，2002.
8. 全国文献工作标准化技术委员会第六分委员会[S]．GB 6447—86 文摘编写规则．北京：中国标准出版社，1986.
9. 王明亮. 关于中国学术期刊标准化数据库系统工程的进展[EB/01]. http://www.cajcd.cn/pub/wml.txt/980810-2.html, 1998-08-16/1998-10-04.

……

附录A ×××××

……

攻读学位期间的研究成果

**已发表论文：**

徐子晨，崔傲，王玉皞，刘韬. 基于RISC-V架构的强化学习容器化方法研究. 计算机工程与科学[J]. 2021，43(2): 70-74.

XXX，XX. 区域经济相互作用研究[J]. 国内经济与中部经济，2002 (已录用).

……

……