

向信创人才培养的信创操作系统实验平台的 建设与探索^{*}

李航 刘高翔 陈星 黄伯虎

西安电子科技大学计算机科学与技术学院, 西安 710126

摘要 针对当前我校缺乏基于信创软硬件的操作系统实验平台的现状,本文在国产泰山 200 服务器和 openEuler 操作系统之上,设计并实现了一种基于 Kubernetes 的操作系统在线实验平台,以助力信创操作系统人才的培养。该平台通过对 Kubevirt 和 Qemu 的定制和适配,实现了在 Kubernetes 集群中远程调试工业级信创操作系统虚拟机的功能,并通过对 Ruby on Rails 的适配,完成了通过 Web 管理 Kubernetes 集群的功能。然后,在该平台的基础上,开发了配套实验案例。经过两个学期的教学实践,该实验平台不但满足了日常教学的需要,而且提高了学生的学习效率并降低了实验平台的维护成本。此外,也发现了一些未来需要解决的问题。

关键字 信创, Kubernetes, Kubevirt, 虚拟机

Building and Investigating an Operating System Platform Focused on Cultivating Talents in the Information and Innovation Sector

Li Hang Liu Gaoxiang Chen Xing Huang Bohu

School of Computer Science and Technology XIDIAN University
Xi'an 710126, China

Abstract—Addressing our school's current lack of an operating system experimental platform for information and innovation (Xin Chuang) hardware and software, this paper designs and implements an online platform based on Kubernetes, utilizing domestic Taishan 200 servers and the openEuler operating system. This platform aims to foster talent in Xin Chuang operating systems. By customizing and adapting Kubevirt and Qemu, it enables remote debugging of industrial-grade Xin Chuang operating system virtual machines within Kubernetes clusters. Furthermore, with the integration of Ruby on Rails, it provides Web-based management of Kubernetes clusters. Then, based on this platform, accompanying experiments are developed. After two semesters of teaching, this platform has not only met daily educational requirements but also enhanced student learning efficiency and reduced maintenance costs. However, some challenges for future resolution have been identified.

Keywords—Xin Chuang, Kubernetes, Kubevirt, Virtual Machine

1 引言

近年来,国际形势风起云涌,为了保障国家安全,中国提出并实施了信创战略。根据 2024 年中国信创产业全景图谱^[1],2029 年信创市场规模将达到 6 万亿元,约为当前的 3 倍。这意味着未来信创产业对人才的需求将极为旺盛。但遗憾的是,当前我国信创人才的培养还存在着不足。以信创产业的关键基石——操作系统的人才培养为例,当前许多高校还缺乏一套完整的基于信创软硬件的实验平台,如实验硬件还是基于国外的 Intel CPU,操作系统使用国外的 Linux 或 Windows 产品,甚至是国外教学用的迷你型操作系统。这将带来如下问题:

(1) 学生难以对信创产品的品质产生信心。如:至少在我校信创实验室的建设实践中表明,我国信创操作系统如麒麟,openEuler,在信创硬件如飞腾服务器和鲲鹏服务器上能够持续可靠地运行,而学生若没有亲自在信创软硬件系统上进行科研或教学实践,对信创硬件和其上信创操作系统的可靠性还是持怀疑态度。

(2) 学生难以深入了解信创操作系统与底层信创硬件之间的交互,这对信创操作系统内核级人才的培养不利。特别是,当前的一些操作系统教学使用的是国外的迷你型操作系统,如: pintos 等,这对信创人才的培养更为不利。这是因为:我们在过去的教学科研实践中发现,虽然这些迷你操作系统能够帮助学生理解操作系统原理,但是即便学生已经熟练掌握迷你操作系统的原理和代码,一旦面临工业级的复杂信

^{*}基金资助: 本文得到西安电子科技大学教育教学改革实践类项目 YQ22005K、ZG-BJ2303、A-XSY2308 的资助

创操作系统上的定制问题,学生依然往往迷失在信创操作系统复杂的代码和控制流中。因为工业级的复杂信创操作系统开发定制对人才的要求侧重点不是掌握较为简单的操作系统原理,而是对复杂软硬件系统的综合理解能力和掌控能力。

针对上述问题,我们提出并设计了一套基于信创软硬件的操作系统在线实验平台。该实验平台设计基于如下两个原则:

(1) 操作系统实验必须运行在虚拟化硬件平台之上

这是因为,前期我们和飞腾的合作中,我们发现若使用硬件开发板会带来两个问题:a.学生容易损坏开发板,使得实验室维护成本急剧升高;b.学生不易观测软硬件行为的交互。而这两个缺点完全可以通过虚拟化解决。此外,虚拟化技术是当前信创操作系统头部企业开发、调试观测操作系统行为的主要工具。实验平台虚拟化技术的引入,能够帮助学生提早掌握信创操作系统开发的主流技术。

(2) 操作系统内核实验必须使用工业级开源信创操作系统,如:openKylin,openEuler,OpenHarmony等,而非采用 pintos 这种教学用操作系统

这是为了训练学生对复杂软硬件系统的理解掌握能力,以应对信创产业对人才的需要。此外,需要指出的是当前的一些信创操作系统因种种原因还未开源,如:银河麒麟,因此像这种闭源信创操作系统,我们未将其作为操作系统内核实验的实验对象。但为了让学生体验更多真实的信创操作系统生态,对于这种闭源信创操作系统,我们将一些操作系统应用层的实验放在其上运行。

本文第1章给出引言;第2章给出该实验平台的设计实现所涉及的关键问题和技术;第3章给出基于该实验平台,适合培养信创操作系统人才的实验案例;第4章给出本实验平台在实际测试和使用中的结果和经验;最后第5章中给出结论。

2 关键问题与技术

下面将从三个层次介绍本实验平台涉及的关键问题与技术:

- 虚拟化集群基础设施选型。
- Kubevirt 适配定制。
- Web 前后端开发。

2.1 虚拟化集群基础设施选型

信创操作系统实验分为两类:应用级实验和内核级实验。对于操作系统应用层实验需要放在容器中运行,以节约集群资源,并避免各个学生做实验时相互

之间的干扰;而对于使用工业级信创操作系统内核实验,则只能使用虚拟机,无法使用容器。所以,虚拟化集群的基础设施必须能够同时支持虚机和容器。此外,虚拟化集群的基础设施必须开源,并且具有庞大的社区,以满足本实验平台的定制需求。

Kubernetes 是一个生态成熟的主流开源容器集群管理系统。信创操作系统的应用层实验可以作为容器放在 Kubernetes 的 Pod 中运行,而 Kubevirt 是 Kubernetes 的一个插件,允许以容器的方式在 Kubernetes 集群中运行虚拟机。

综合上述因素,本实验平台选择 Kubernetes+Kubevirt 作为虚拟化集群的基础设施。

此外,为了尽可能地减少开发时与信创软硬件的适配问题,本实验平台所在服务器的配置为:硬件选择华为泰山 200 服务器,信创服务器操作系统选择华为 openEuler 操作系统。

2.2 Kubevirt 适配定制

虽然在开发过程中,openEuler 操作系统与泰山 200 服务器并未出现适配问题,但 Kubevirt 和泰山 200 服务器出现了适配问题,即 Kubevirt 在生成虚机时,出现“缺少 Pid”的错误。我们将该问题提交到 Kubevirt 社区^[2],社区开发者通过修改 Kubevirt 源码,并发布了新的 Kubevirt 版本,使得该问题最终得到了解决^[3]。

此外,出于安全考虑,Kubevirt 的官方发布的版本禁止直接以调试模式运行虚机,因此无法满足学生调试信创操作系统内核的实验需要^[4]。为此,需要对 Kubevirt 进行定制。通过对 Kubevirt 静态源码分析和动态运行分析,我们发现 Kubevirt 运行虚机的本质是将 libvirtd 和 Qemu 包装为一个 Pod,并在该 Pod 中通过 libvirtd 启动 Qemu,并且以 CRD 的形式将虚拟机管理接口接入到 Kubernetes 中。而 Kubevirt 在启动 Qemu 时,并没有向 Qemu 中传递调试参数,因此 Kubevirt 并没有以调试模式启动虚机。

基于上面认识,我们定位了 Kubevirt 中启动虚机的源码行,并给其增添了“-s”调试参数,并重新编译,最终成功使得 Kubevirt 以调试模式运行了虚机。基于这样修改过的 Kubevirt 创建虚拟机之后,gdb server 会监听虚拟机所在 Pod 的 1234 端口,gdb 客户端只需通过远程调试方式访问这个端口就可以调试虚拟机。

此外,为了避免恶意用户绕过本实验平台的认证机制,直连 gdb server 所监听的端口或直连虚拟机端口,本实验平台采取了如下策略:

在创建管理虚拟机的 Pod 时,同时一并创建运行

gdb 客户端的 Pod^[5] (以下简称客户端 Pod), 以及该客户端 Pod 的 NodePort 类型的 Service 资源。该 Service 用于暴露客户端 Pod 的 22 端口至物理服务器, 使得用户能够通过远程 SSH 连接进入客户端 Pod。当用户进入到客户端 Pod 中, 由于客户端 Pod 和管理虚拟机的 Pod 处于同一个 Kubernetes 集群当中, 客户端 Pod 可以直接通过虚拟机在 Kubernetes 集群内部的 ip 以及虚拟机本身的端口号访问虚拟机。同时要创建 NetworkPolicy, 用于配置网络策略^[6], 通过 label 标签将创建的客户端 Pod 和管理虚拟机的 Pod 绑定在一起, 只允许客户端 Pod 访问该虚拟机, 无法通过其它 Pod 访问这个虚拟机。

2.3 Kubernetes Web 管理页面开发

本实验平台使用 Ruby Rails 框架开发 Kubernetes 的 Web 管理页面。在实际中我们发现, Ruby Rails 在安装 openEuler 操作系统的 x86 架构的开发机上进行部署时, 没有适配问题, 而在实际运行环境, 即在信创泰山 200 服务器与 openEuler 操作系统构成的服务器环境下部署时, 会出现适配问题。具体表现如下: 在实际运行环境中, 虽然能够安装 Ruby Rails, 但只要通过 Ruby Rails 删除 Kubernetes 集群中的虚拟机就会出现物理服务器上的 openEuler 操作系统内核错误并导致物理服务器自动重启。经过排查后发现, 如果在物理服务器上安装了 Ruby 语言的环境, 就会导致这个问题, 若不安装该环境反而可以正常删除虚拟机。目前认为可能的原因是: 在安装 Ruby 的过程中, 除了需要通过 make 命令对 Ruby 的源代码进行编译, 同时还需要安装相关的编译依赖, 可能这个过程当中的某一步, 如某个依赖包, 会对操作系统内核造成影响; 或者 Ruby 安装文件与 Kubernetes 的安装文件产生了冲突。基于上述猜测, 我们通过在 Host 机上的 libvirt 创建一台独立于 Kubernetes 的虚拟机, 并将 Ruby Rails 和 Web 应用部署到该虚拟机中, 使得 Host 机与 Ruby Rails 环境相隔离。最终成功解决了该问题。

此外, 在开发过程中, 我们也发现, Kubernetes 的官方文档也有不足之处, 如: 查看 Kubernetes API 和 Kubevirt API 说明可以发现, 其中关于资源删除失败时的响应示例只有 “Unauthorized”, 也就是没有权限。但在实际测试中, 资源删除失败还有其他可能性, 如: 当 Kubernetes 集群有请求无法及时处理时将会返回 “Internal error” 网络错误。因此为了保证资源能全部删除, 在删除操作中, 采用循环发送删除请求的方式, 即每发送一次删除请求, 都判断是否删除成功, 如果不成功则重新发送删除请求, 直至删除成功。

最后一项在 Kubernetes Web 管理页面开发中用到的关键技术是资源回收技术。这是为了做到定期释放

资源, 以防止学生实验完毕之后, 忘记释放虚拟机资源的情况发生。该功能需要使用异步通信机制来实现。异步通信机制的实现方式有很多种, 如果依靠手动创建线程来实现, 则必须考虑异常处理和安全等问题, 容易出错; 如果使用第三方的消息队列或者排队后端来实现, 则可能又会引发适配问题。因此经过综合考虑, 对于这一功能的实现, 我们选择使用 Ruby Rails 中的 ActiveJob 排队作业来实现。ActiveJob 是一个声明作业并使其在各种排队后端上运行的框架。这些作业可以是定期计划的清理或邮件发送等任务。该方法的缺点是: 与第三方排队中间件相比, ActiveJob 排队作业将任务保存在内存中。因此当进程崩溃或机器重启时, 未完成的任务会丢失。不过在本项目中, ActiveJob 排队作业的任务丢失在容忍范围之内。

3 实验案例

基于上面的实验平台, 结合信创产业对人才的实际需求, 我们设计了相关实验案例。这些实验案例涵盖了信创操作系统开发的主要关键技术: 从内核基础 API 以及数据结构的使用, 到硬件设备驱动的编写, 最后使用内核和根文件系统裁剪技术, 搭建一个自定义的操作系统。

3.1 案例设计原则

由于信创操作系统的开发往往更多接触到操作系统底层相关的知识, 这要求学生操作系统内部机制有深入的了解, 而另一方面, 信创操作系统的软硬件高度耦合, 构成了一个极为复杂的系统, 最终造成学生学习理解信创操作系统的高门槛。前期, 我们在信创头部企业的调研也证实了这一点。因此如何让学生在有限的实验时间中迅速掌握信创操作系统的开发技巧, 是对信创操作系统教学新的挑战。

针对该问题, 我们在设计该案例时, 遵循 “由表及里” 的设计理念, 着重培养学生层次化理解复杂系统工程问题能力。具体如下:

实验案例共分为两大类: 应用级实验和内核级实验。

应用级别实验以 Shell 脚本编写为起始, 接着, 通过让使用系统调用追踪工具, 观察 Shell 脚本执行时所使用的系统调用, 然后让学生使用系统调用进行编程, 最后通过让学生在内核中添加系统调用, 让学生进入内核级实验。通过上述过程, 学生能够亲身切实感受到操作系统应用层中各个子层之间的关联关系, 形成自己对操作系统应用层中各个子层的知识理解体系, 并且认识到应用级实验和内核级实验之间的关系。

对于内核级实验, 依然采用同样的方式进行设计, 即从内核最基础的 API 和数据结构开始, 逐步深入到

硬件接口编程。具体如下:

- ① 内核线程使用与内核链表操控。
- ② Deferred work。
- ③ Edu 设备驱动。
- ④ 内核与根文件系统裁剪。

其中,实验2是对实验1的扩展;实验3需要用到1或2技术,并需要和硬件进行交互;实验4是综合上述3个实验相关技术的系统性实验,它是对信创企业中常见的嵌入式系统移植实践的抽象。内核级的具体实验内容如下所示:

3.2 内核线程使用与内核链表操控

通过本实验的学习,学生需要掌握信创操作系统内核定制中所常用的内核数据结构和函数,具体包括:

- 内核链表。
- 内核内存的分配释放。
- 内核线程生成与释放。
- 内核锁。

实验内容如下:

设计一个内核模块,并在此内核模块中创建一个生产者线程和一个消费者线程。其中,生产者线程需要遍历进程PCB链表,并将PCB链表上各个进程的相关信息,如:pid、进程名等加入到自己所设置的内核链表中;消费者中需不断从该内核链表中取出相关信息。最后,在卸载模块时停止内核线程并释放资源。

该实验在实施时,需要重点考察学生对内核级编程和应用级编程的区别,如:内核线程和应用级线程在使用上的不同之处等,以帮助学生尽快适应内核开发环境,并在头脑中与建立与已有应用级编程知识点的关联关系。

3.3 deferred work

通过本实验的学习,学生需要掌握信创操作系统内核定制中的workqueue技术,理解其与kernel thread的区别。实验内容如下:

设计并实现一个内核模块,该模块旨在通过workqueue和kernel thread两种不同的机制来完成内核任务,并对比分析这两种实现方式的异同点。

该实验在实施时,需要重点考察学生对workqueue技术出现的背景,特别是其所针对问题的理解,进一步帮助学生理解内核在处理任务时的独特之处。

3.4 PCI 设备驱动

通过本实验的学习,学生需要掌握信创操作系统

内核定制中所常见PCI设备驱动适配技术。本次实验旨在让学生深入理解并实践信创适配中的PCI设备驱动的开发技术。不过,由于PCI设备驱动较为复杂,为了帮助学生快速理解PCI驱动开发中的关键要素,实验中,我们提供了PCI设备驱动的框架代码,学生在该框架代码上进行填空式编程。这样减少非重要性细节对学生的干扰,并且也符合信创业界开发时的常见场景。该实验要求学生掌握的重点技术点如下:

- ① PCI设备私有数据成员空间的分配与释放;总线地址与虚地址的转换;PCI设备使用时原子性的保证等。
- ② PCI设备驱动和操作系统内核的对接以及PCI设备驱动与PCI设备的对接。
- ③ PCI设备驱动识别不同进程调用的能力,确保将计算结果正确返回给对应的调用进程。
- ④ 应用级程序对PCI设备驱动的访问。

该实验在实施时,需要重点考察学生对PCI驱动中不同的控制流的理解,特别是应用层的程序时如何最终访问到该PCI硬件的关键路径。这是为了帮助学生建立对由应用层、内核层以及硬件层构成的复杂系统的贯穿式理解。

3.5 内核裁剪

通过本实验的学习,学生需要掌握信创操作系统内核裁剪、根文件系统定制以及内核调参技术。在本次实验中,学生将利用qemu模拟器启动并运行一个自定义的由内核与上层应用构成的完整操作系统。该实验要求学生掌握的重点技术点如下:

- ① 根据定制目标,对信创操作系统内核进行配置编译,以使操作系统内核在满足定制目标的要求下,能够以尽可能小的体积运行。
- ② 根据定制目标,能够配置编译生成与上述内核相匹配的根文件系统。
- ③ 给自己定制的操作系统中添加“PCI设备驱动”实验中已经编译完成的PCI设备驱动,并进行安装和运行测试。
- ④ 启动自己定制的操作系统后,修改指定内核参数的值。

该实验在实施时,需要重点考察学生对内核与根文件系统匹配的理解程度,以及上述技术点在信创操作系统开发中的应用场景。这样做除了帮助学生建立对复杂系统的贯穿式理解外,更重要的是帮助学生了解信创操作系统开发产业中的技术基础概貌,奠定学生提前思考职业规划的技术基础。

4 结果分析

本实验平台已在实际教学中投入使用,其部署环

境如下:

两台信创泰山 200 服务器,服务器的硬件配置为:CPU 共两颗(每颗 48 核,共 96 核,频率 2.6G),型号为鲲鹏 920-4826,内存 192G,4T 硬盘。宿主机和虚拟机均运行的是 openEuler-22.03-LTS-aarch64 操作系统。每台虚拟机的配置都为 2 核 2G。

经过先期两个学期的实验班学生(1 个班为 21 人,另一班为 22 人)的操作系统课程实践,我们积累了如下经验:

(1) 内核编译时长

实际使用中得到的统计结果如表 1 所示。该统计结果所针对的统计对象是操作系统的系统调用添加实验,该实验是典型的操作系统内核级实验,需要通过虚拟机而非容器才能完成。

表 1 给出了 1/2/15 台虚拟机同时编译内核的相关数据。在实际使用中,由于学生做实验的速度差异以及错误修复所耗时间的不同,通常不会出现满负荷使用全部实验虚拟机进行编译内核的情况。经实践检验,同时使用 15 台虚拟机进行编译是我们观测到的典型情况。从表 1 中可以看出,15 台虚拟机同时编译内核的时长约为 1 台虚拟机编译时长的 2 倍。此外,15 台虚拟机同时编译时,I/O 的带宽占用率已经接近 100%;而 1 台虚拟机编译时,I/O 带宽占用率仅为 4.53%。

基于上述实验数据可以判断,I/O 磁盘带宽不足是编译时长翻倍的原因,也是当前平台的最先遇到的瓶颈。

表 1 内核编译性能统计

同时编译内核虚拟机数量	1个	2个	15个
内核平均编译时间	31.3分钟	37.1分钟	58.6分钟
CPU占用率	2.1%	4.2%	30.2%
I/O磁盘带宽占用率	4.53%	4.57%	99.8%

(2) 实验环境搭建成本的降低

本实验平台的引入,大大降低了实验环境搭建成本。学生可以直接使用现成的实验环境,而不需要手动搭建。以往即使在网盘中提供了虚拟机镜像下载,也会有约 10%的学生在系统安装时出现了问题,需要老师介入指导,这拖慢了学生的学习过程。

(3) 硬件开发板成本的节省

针对操作系统 I/O 驱动程序实验,我们使用 Qemu 对 I/O 硬件进行了模拟,因此,学生在进行 I/O 驱动程序实验时,只需在虚拟机中完成,无需再使用硬件开发板。而往常硬件开发板在每次课程结束后,约有 15%的损坏率,需要花费额外的成本进行维修或购置。

5 结束语

本文从信创操作系统人才培养角度出发,基于当前国内高校缺乏以信创软硬件为基础的在线实验平台的现状,设计并实现了一套以鲲鹏 920 处理器和 openEuler 操作系统为基础设施,以 K8s+Kubevirt+Qemu 为上层应用,并搭配 Web 管理界面的在线信创操作系统实验系统。在此基础上,设计并实施了一套系统化的信创操作系统实验案例,涵盖信创操作系统开发过程中所涉及的关键知识点,以帮助学生了解并掌握信创操作系统产业中的相关技术,奠定其信创职业规划的技术基础。

另外,本实验平台具备高度可扩展性。这是因为本实验平台提供了完整的虚拟机调试环境和完整的开发工具链,并且借助 Qemu 的设备模拟功能可以实现对大多数常见设备驱动的测试工作。用户只需要根据自身需求,修改实验案例,即可完成所需的与操作系统相关的开发任务。

通过实践检验,本实验平台已满足了本学院信创操作系统实验教学的日常需要,提高了学生的学习效率,降低了实验平台的维护成本。此外,上述实践也表明在国产信创硬件和信创操作系统上建设信创操作系统实验平台是可行的,但还需解决一些适配问题。这些适配问题可以通过与开源社区紧密协作,或者自行定制,或者通过虚拟机的方式进行解决。

在 2024 年秋季学期,我们首批购入 4 台新的信创服务器,以扩大本实验平台的规模,并且,后期将追加购入 10 台以上的信创服务器,以使本实验平台向全校开放。不过,需要指出的是,在实验平台前期开发过程中,我们发现有人恶意利用本实验平台的计算资源,运行挖矿程序。为此,我们已完成了一套虚拟机监控工具,计划在 2024 年底投入使用。

参考文献

- [1] 前瞻产业研究院,2024 年中国信创产业全景图谱,[2024-5]
<https://bg.qianzhan.com/report/detail/2009211055112507.html>
- [2] 刘佳明,李航等,Bug Report,[2022-4-21]
<https://github.com/kubevirt/kubevirt/issues/7603>
- [3] Jed Lejosne, Patch Report,[2022-4-27]
<https://github.com/kubevirt/kubevirt/pull/7637>
- [4] Kubevirt API reference. [EB/OL].[2022-04].
<https://kubevirt.io/api-reference/main/definitions.html#:~:text=Type%20%3A%20object,-v1.SEV,-Name>
- [5] 陈丰琴. 基于 Kubernetes 集群容器资源调度策略的研究与设计[D]. 西南交通大学,2020
- [6] Budigiri, Gerald et al. "Network Policies in Kubernetes: Performance Evaluation and Security Analysis." 2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit) (2021): 407-412.