内核热升级介绍

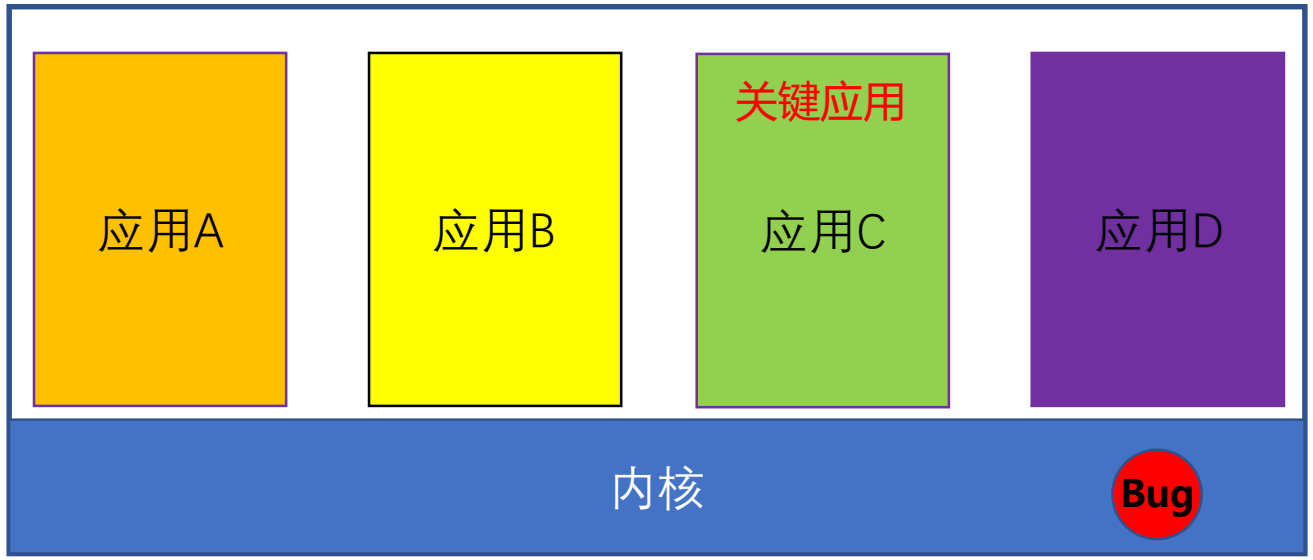
庞加莱

桑琰

2022.10.11

目录

内核热升级简介



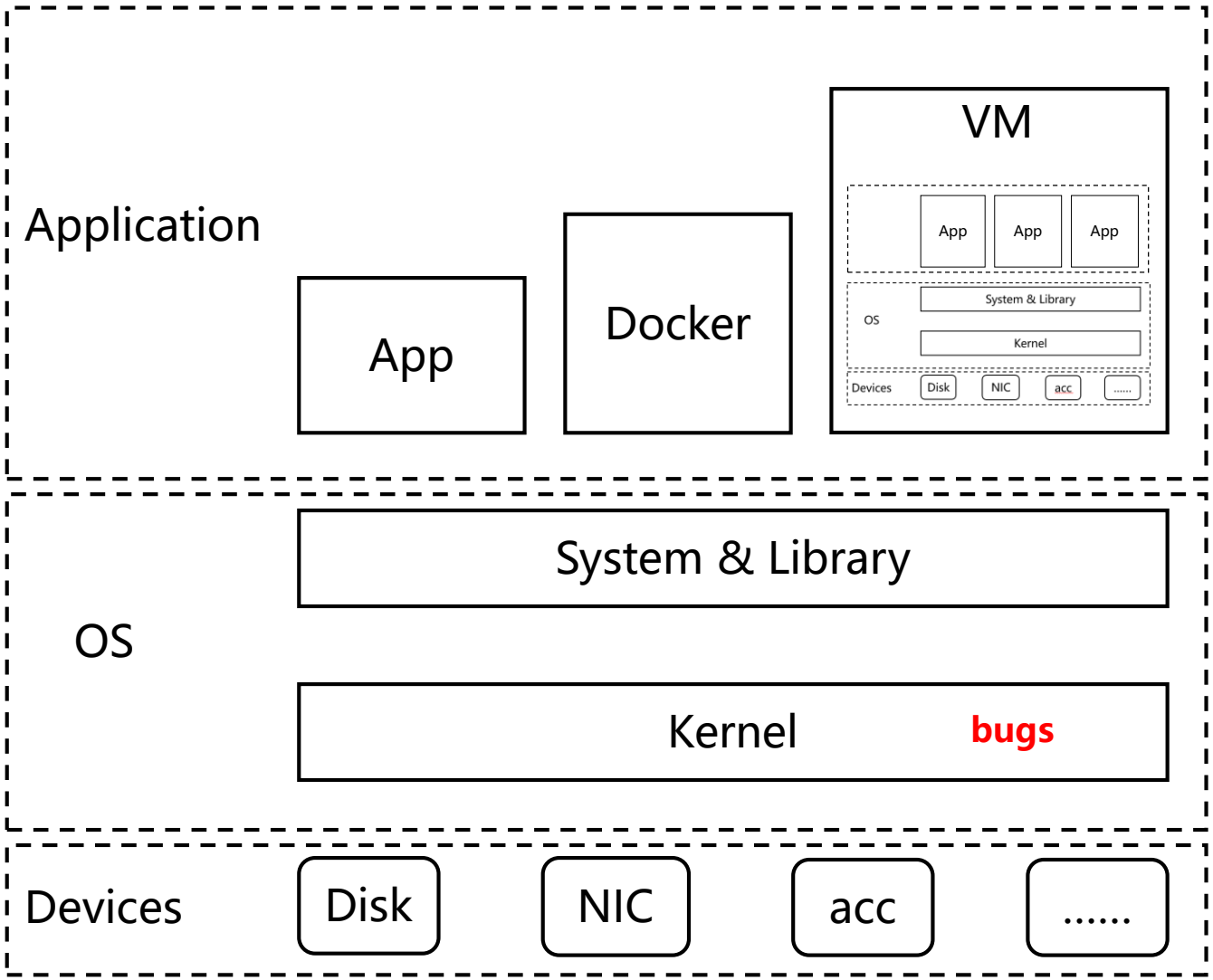
KEXEC + CRIU

内核bug修复 + 关键应用不中断

- Kexec机制: 快速启动新内核，实现内核快速升级；
- Criu机制：内核快速升级过程中，备份恢复关键应用状态，实现关键应用状态不丢失。

目录

内核热升级背景意义-业界研究现状



Kernel Upgrade and Bugs Fix Methods

➢ 内核热补丁

1、热补丁无法修复所有内核缺陷

热补丁无法修复的内核缺陷主要包括如下几类：

1）缺陷修复代码中存在结构体改变；

2）缺陷修复代码中函数参数发生改变；

3）缺陷修复代码中存在内联函数；

4）缺陷修复代码逻辑功能发生大改变，代码框架结构变更。

2、热补丁管理成本高，源码溯源困难

热补丁数目随着漏洞数目线性增长，版本维护困难。

➢ APP/VM 热迁移

热迁移机制存在如下难以克服的问题：

1）无法解决硬件设备直通问题

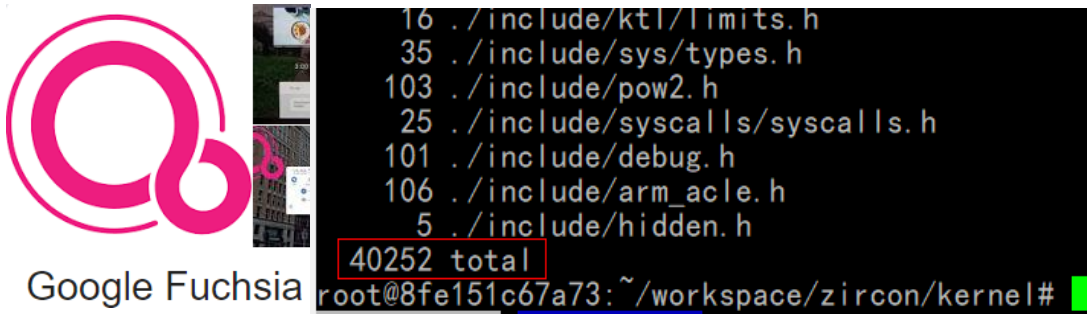
2）大容量内存数据传输困难

➢ 内核热升级

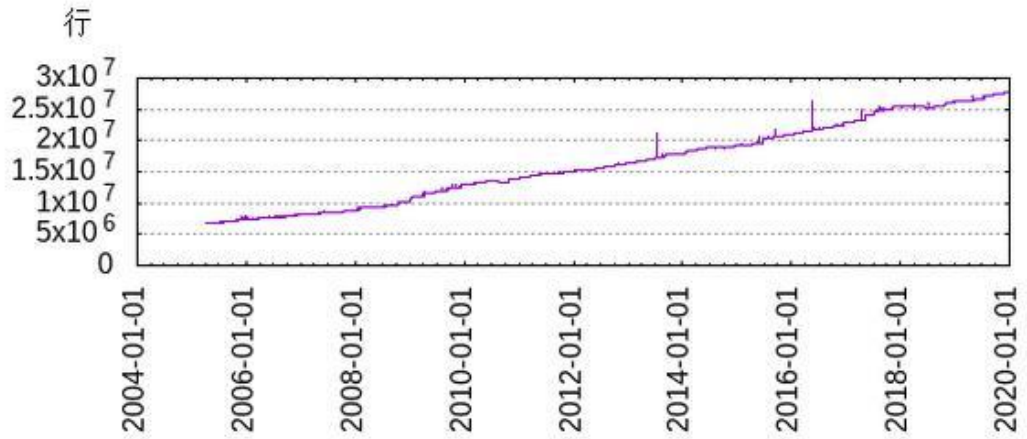
业界暂无公开的成熟方案

内核热升级背景意义

- 微内核规模 -- 规模小，功能单一，易于序列化/反序列化



- 宏内核规模 -- 规模巨大，序列化/反序列化困难

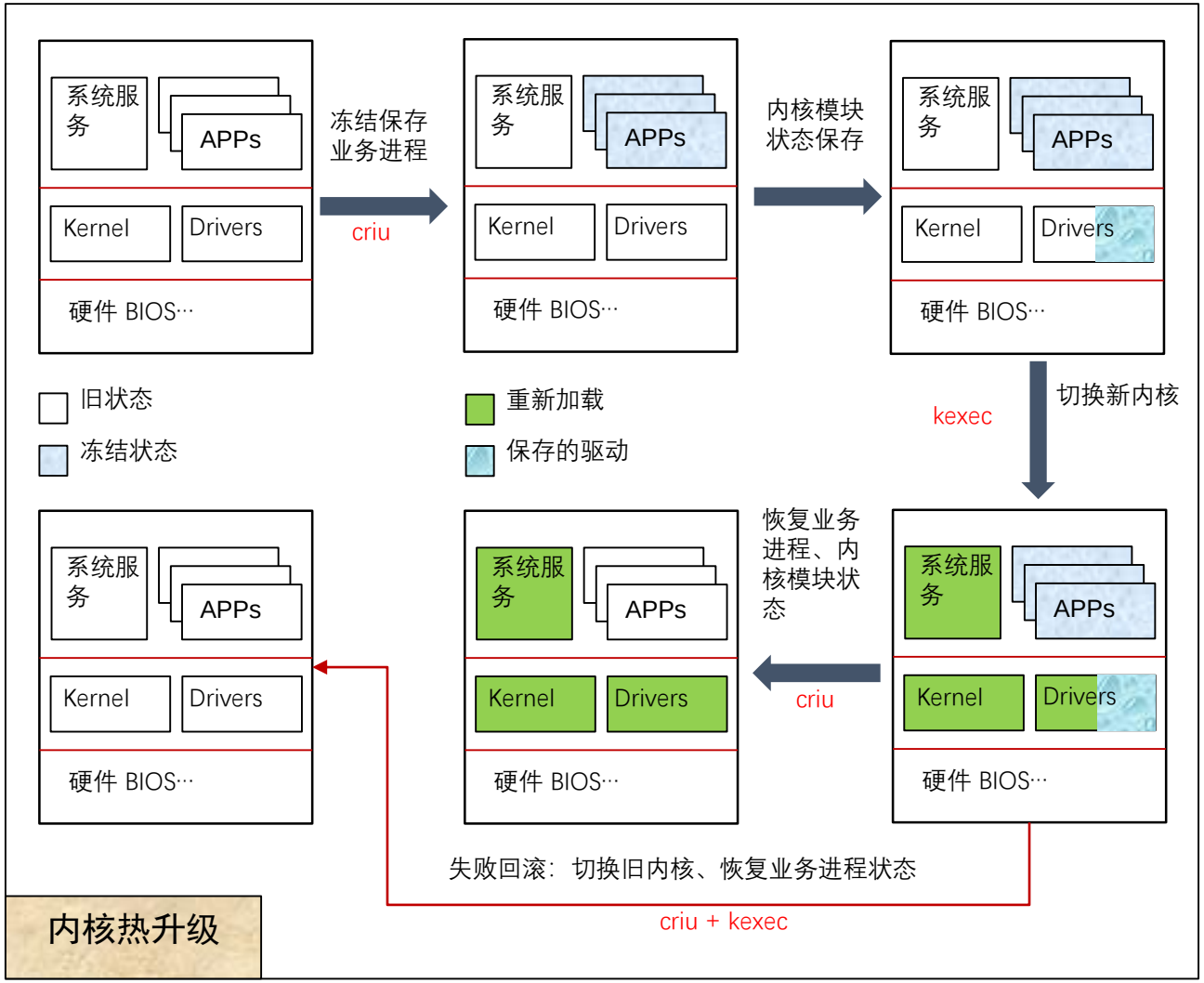


Linux内核源代码包括文档，Kconfig文件，用户空间实用程序等，共有两千七百万行

微内核易于实现热升级，宏内核较难实现热升级，同时，宏内核更容易出问题，内核热升级机制对linux宏内核意义重大

目录

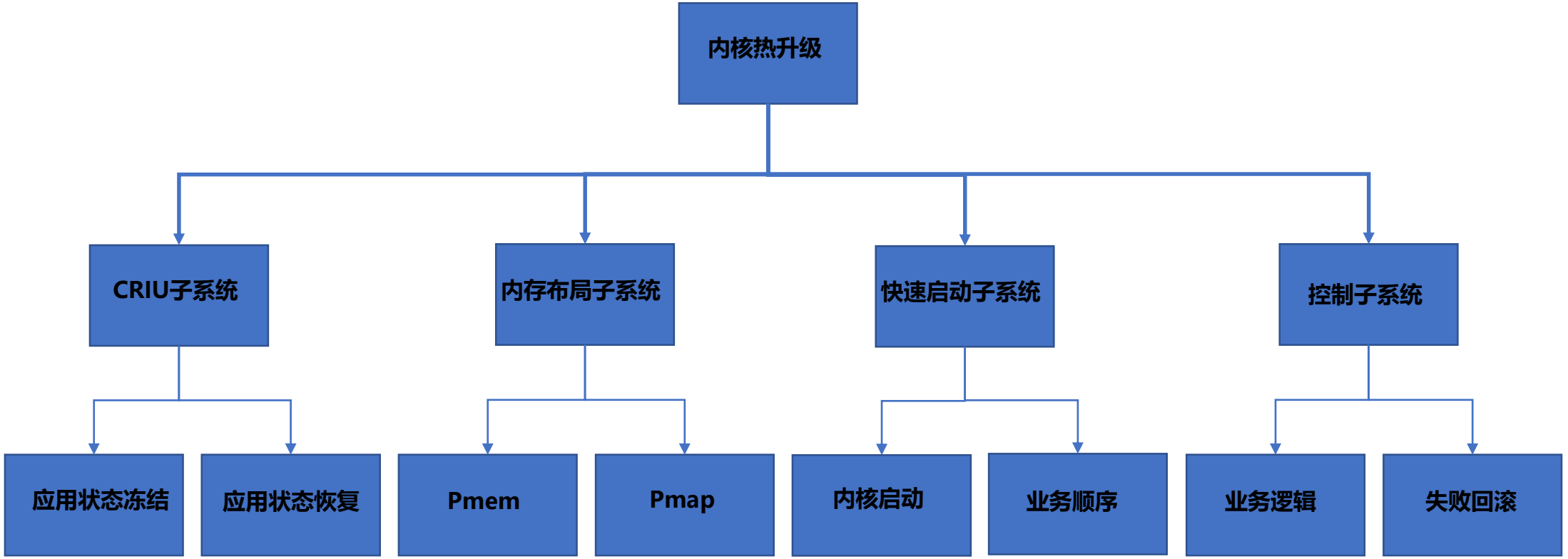
内核热升级架构-原理框图



关键技术：

1. **应用状态备份恢复技术**：将进程或虚拟机的完整状态冻结在内存中，内核切换后进行进程/虚拟机的恢复，用户态无感知；
2. **内核快速重启技术**：社区 kexec基础上增加优化，使内核在500ms内完成重启切换；
3. **硬件状态保持技术**：硬件、BIOS等不进行操作，保持状态不变，内核替换过程中DMA等可以继续运行。

内核热升级架构-组件框图

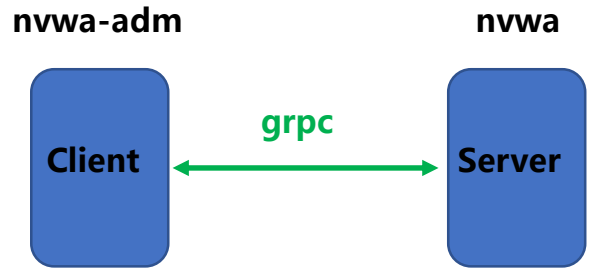


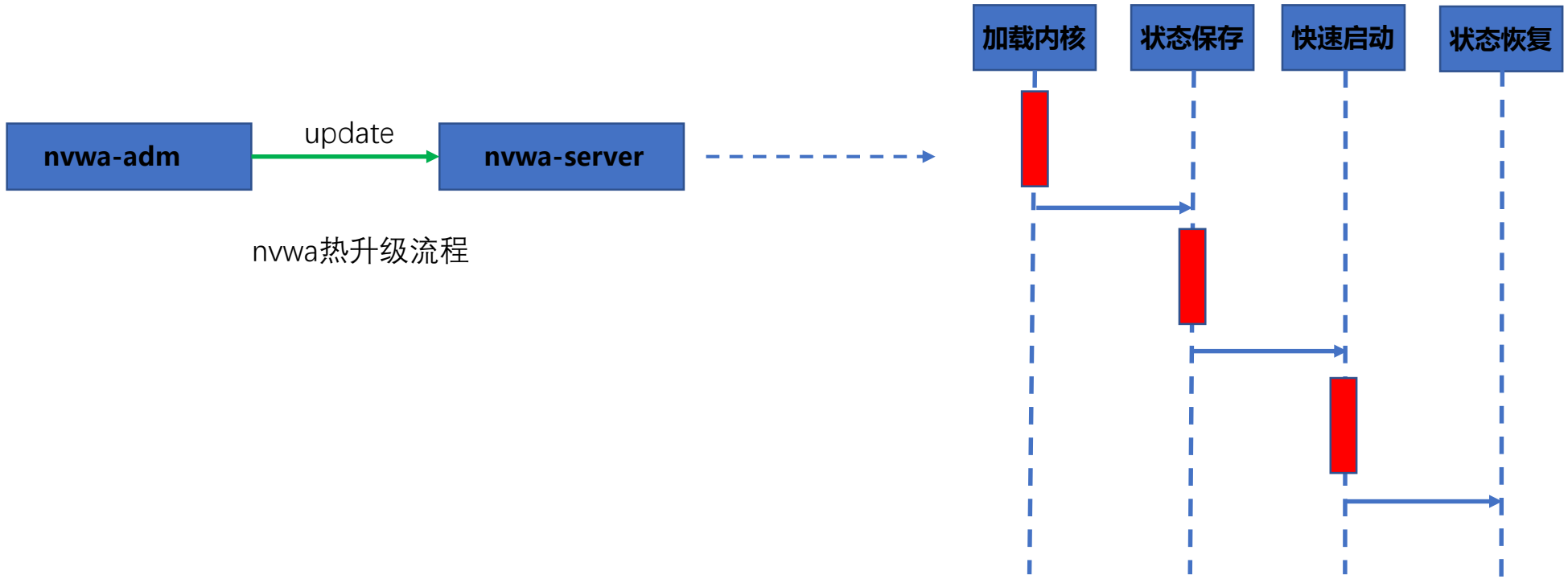
目录

内核热升级-控制子系统

nvwa-adm 客户端命令：

- nvwa-adm update ${version}





nvwa热升级流程

内核热升级-快速启动子系统

Kexec快速启动内核

Kexec目标：
跳过BIOS/GRUB阶段，减少内核重启时间。

kexec整体思路如下:
1）新的kernel镜像和initrd镜像连续存储在内存中，initrd的位置记录在boot_params中；
2）切换到新内核就是跳转到新的kernel镜像所在内存位置，CPU执行其entry的代码即可，新的内核通过boot_params记录的initrd位置完成根文件系统内容的加载。

瓶颈点:
新内核加载的目标位置正在被当前内核使用，需要将内核启动文件先分散存储在能申请到的内存页面中，在新内核跳转前搬移拼接到目的地址。

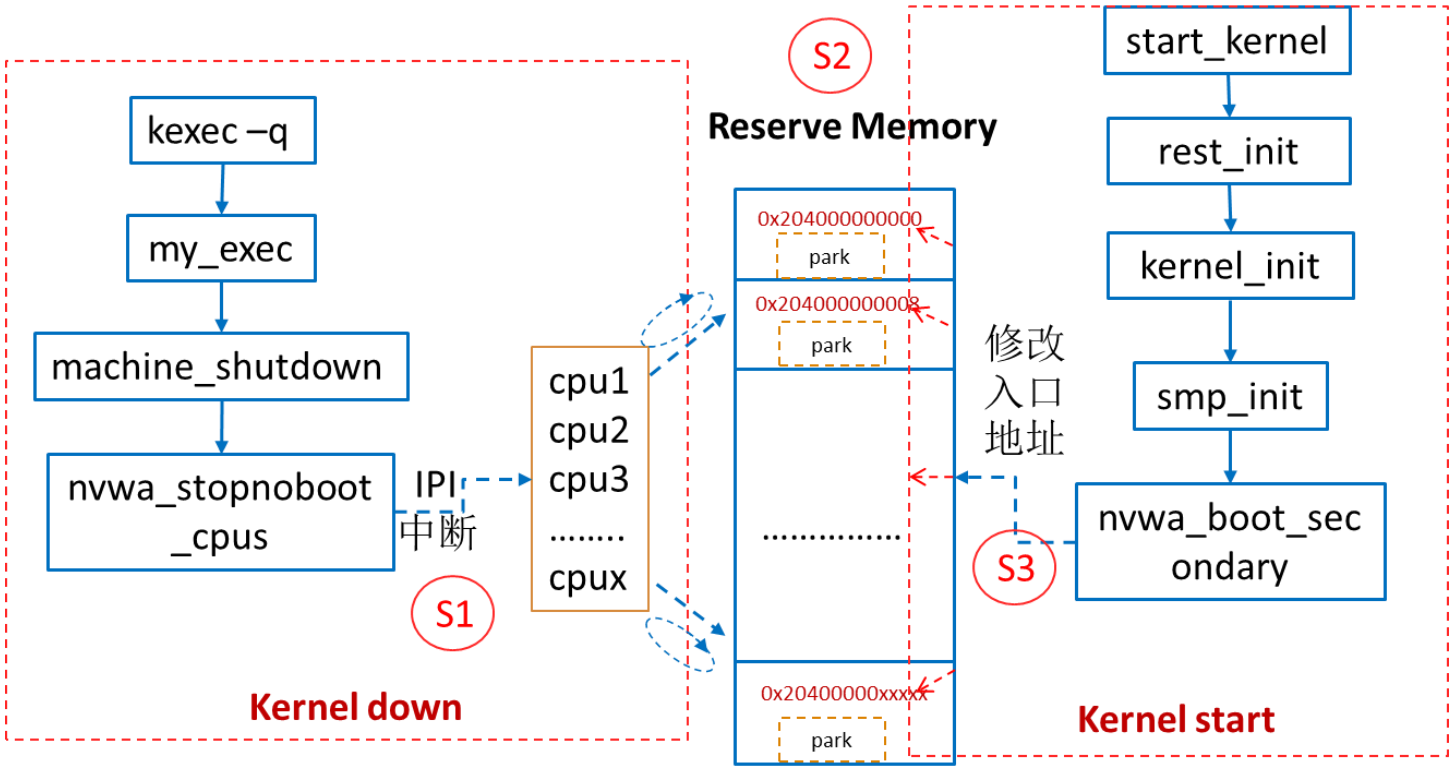
优化方法：
预留内存给新内核加载，基于该预留内存规划新内核启动文件加载位置，避免内存数据搬移拼接。

内核热升级-快速启动子系统

CPU快速启动机制-CPU park/unpark

基本原理：

cpu不用power off和on，直接跳转到新内核地址进行初始化。



Park/Unpark：cpu在park状态时持续check reserve memory中的值是否为0，不为0跳入该入口地址，进入Unpark状态

基本流程：

S1：Kernel down

执行Kexec –e之后，cpu0给其他cpu发送IPI中断进入park状态

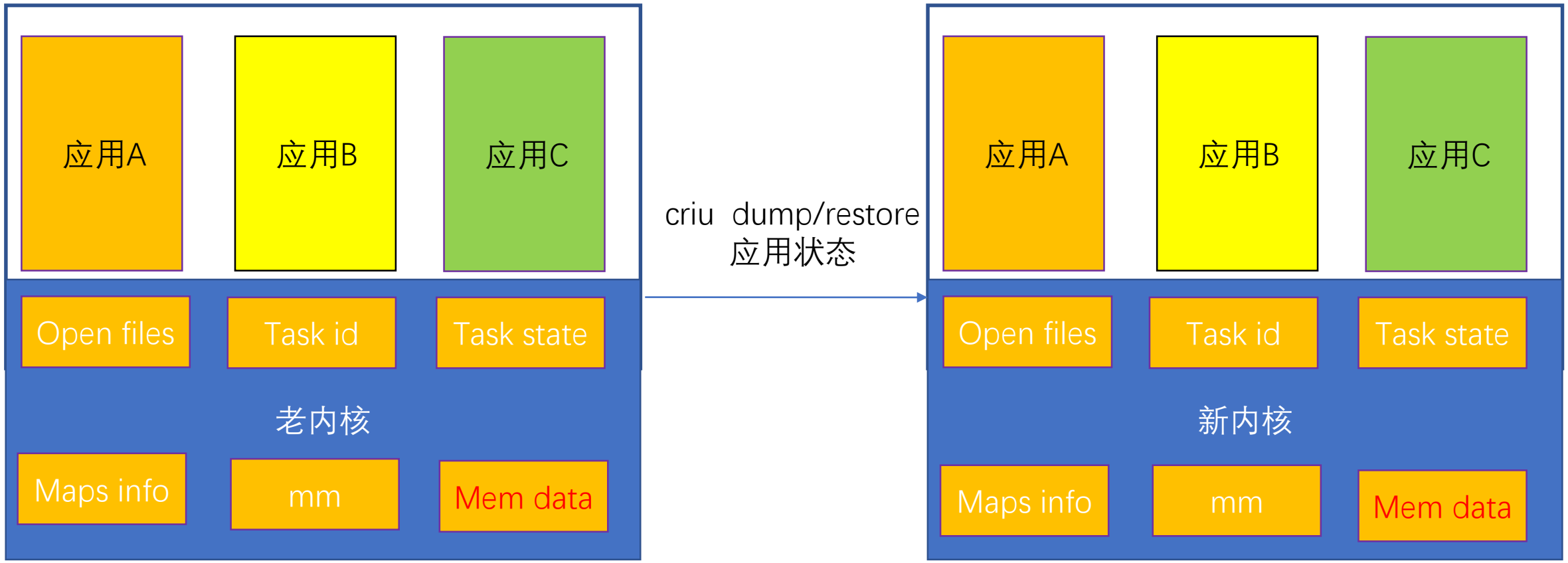
S2：预留内存

预留内存来存储cpu的入口地址和park状态代码段

S3：Kernel up

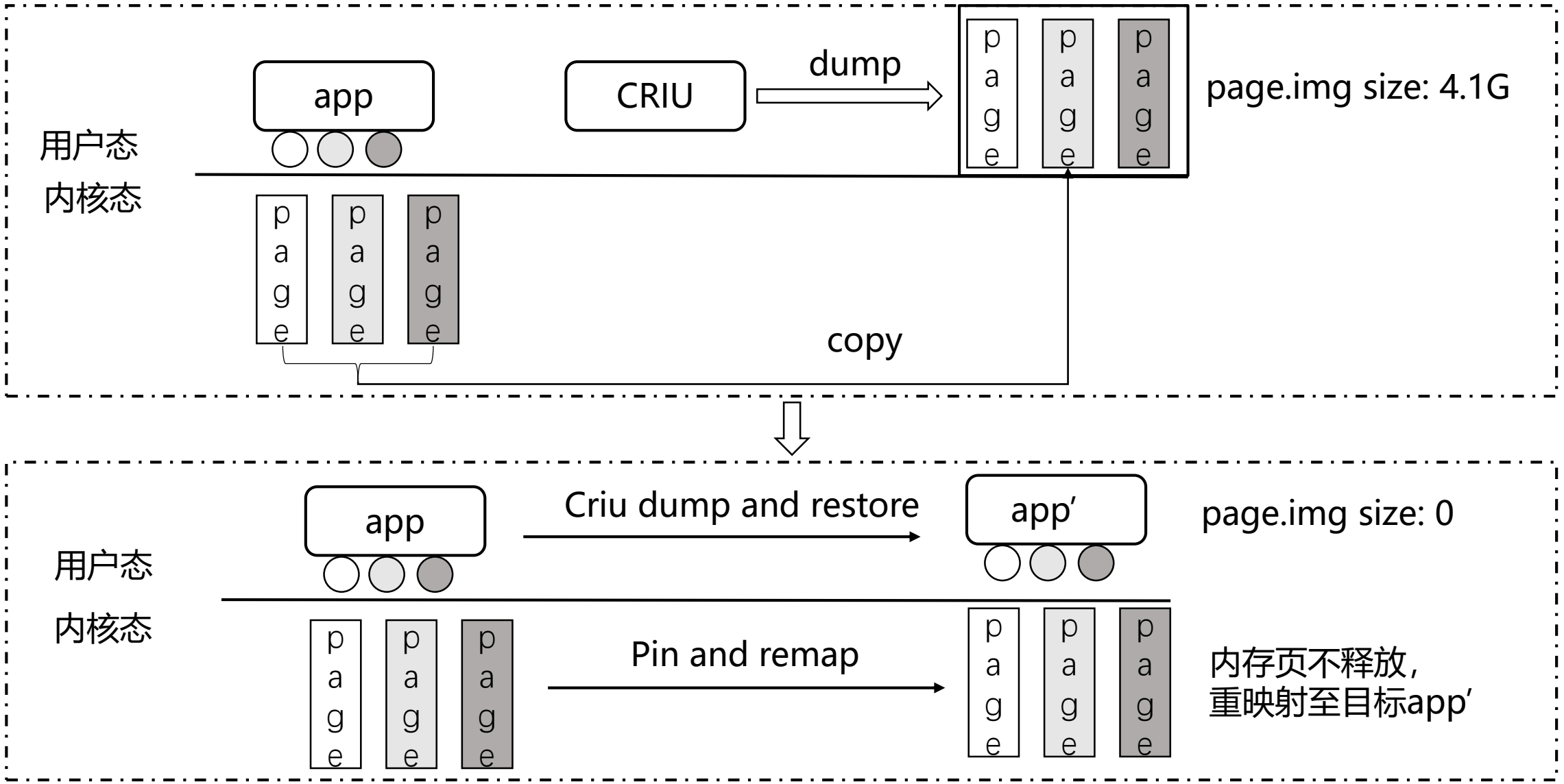
start kernel后，nvwa相关函数修改各cpu的入口地址，cpu检测到后，进入unpark状态。

内核热升级-CRIU子系统



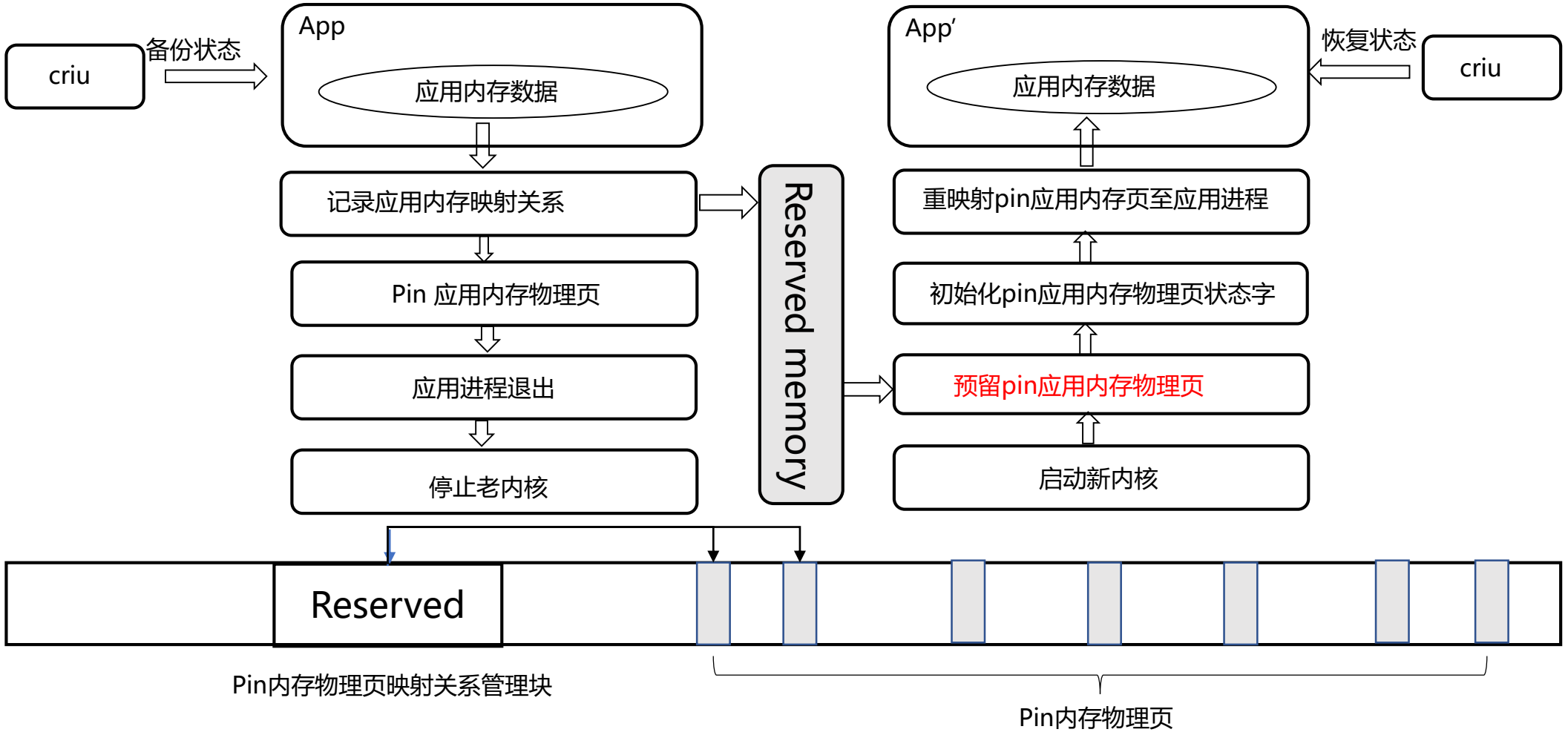
难点：应用内存数据量大，采用开源数据备份恢复机制耗时长，无法满足业务不中断需求，且有信息泄露风险！

内核热升级-CRIU子系统



内核热升级-CRIU子系统

应用内存快速恢复机制



目录

内核热升级管理包安装

官方文档

内核热升级官方文档地址：
https://docs.openeuler.org/zh/docs/21.03/docs/KernelLiveUpgrade/KernelLiveUpgrade.html

更详细的使用方法和FAQ可以在论坛查看：

https://forum.openeuler.org/t/topic/66

组件安装

[root@openeuler ~]# yum install -y nvwa criu kexec-tools

内核热升级安装文档：xxx

编译安装

环境准备：安装 go

参考链接： https://gitee.com/src-openeuler/nvwa?_from=gitee_search

获取nvwa源代码：git clone https://gitee.com/src-openeuler/nvwa.git

编译nvwa：cd src; go get nvwa; go build

内核热升级运行方法

1. 准备运行环境

安装软件包：yum install -y nvwa criu kexec-tools

向内核添加启动参数：quickkexec=350M max_pin_pid_num=4096 pinmemory=100M@0x64000000

cpuparkmem=0x40000000

向内核加载必要的模块：modprobe pin_memory

2. 获取升级版本内核rpm包并安装

从openEuler获取内核rpm包： wget https://repo.openeuler.org/openEuler-22.03-LTS/everything/aarch64/Packages/kernel-5.10.0-60.18.0.50.oe2203.aarch64.rpm

安装内核rpm包：rpm -ivh kernel-5.10.0-60.18.0.50.oe2203.aarch64.rpm

3. 使用命令行启动

nvwa check：运行环境检查
nvwa update：热升级到相应的内核版本(相关文件需放置在/boot下)
nvwa将会去/boot目录下寻找需要的kernel和rootfs，kernel的命名格式需为vmlinuz-${version}，rootfs命名格式需为initramfs-${version}.img
nvwa restore：恢复某个之前freeze的进程
nvwa help：
显示client相关的帮助信息
nvwa --help：
显示server相关的帮助信息

相关patch链接

In kernel

Quick kexec:

http://patchwork.huawei.com/patch/115014/

http://patchwork.huawei.com/patch/115015/

Cpu Park:

http://patchwork.huawei.com/patch/115016/

Pmem

http://patchwork.huawei.com/patch/115017/

Pin mem

http://patchwork.huawei.com/patch/115018/

Pid reserve

http://patchwork.huawei.com/patch/115019/

Criu

criu: A tool of Checkpoint/Restore in User-space - Gitee.com

0001 和 0003两个patch

Kexec-tools

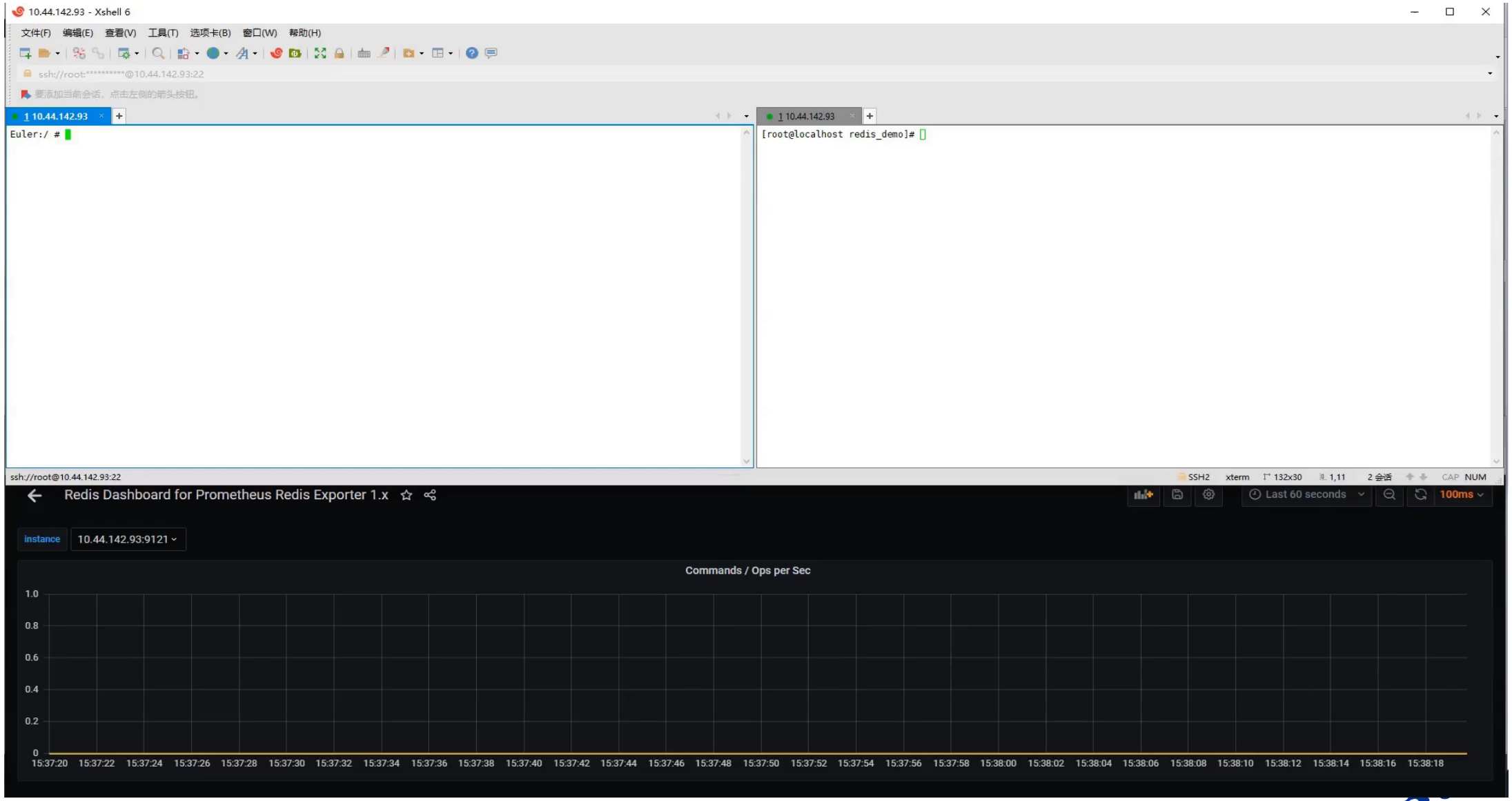
https://gitee.com/src-openeuler/kexec-tools/blob/openEuler-22.03-LTS-SP1/kexec-Add-quick-kexec-support.patch

https://gitee.com/src-openeuler/kexec-tools/blob/openEuler-22.03-LTS-SP1/kexec-Quick-kexec-implementation-for-arm64.patch

其他未合入的优化patch

https://gitee.com/openeuler/nvwa/tree/master/patches/kernel-5.10

内核热升级演示



openEuler


OpenEuler