基于标签化RISC-V架构的进程共享资源管理系统

操作系统 课程设计

最终报告

尤予阳

2021.5.22

目录

设计目标

- 复现标签化RISC-V架构目前的成果
- 将uCore-SMP移植到标签化RISC-V硬件平台上运行
- 在内核中管理进程使用的内存带宽和共享缓存资源
- 设计测例，演示控制产生的效果

已有工作

- 标签化RISC-V架构
- Linux中的Control groups
- Arm v8.5-A 内存标记扩展
- Intel资源调配技术
- uCore-SMP

设计方案

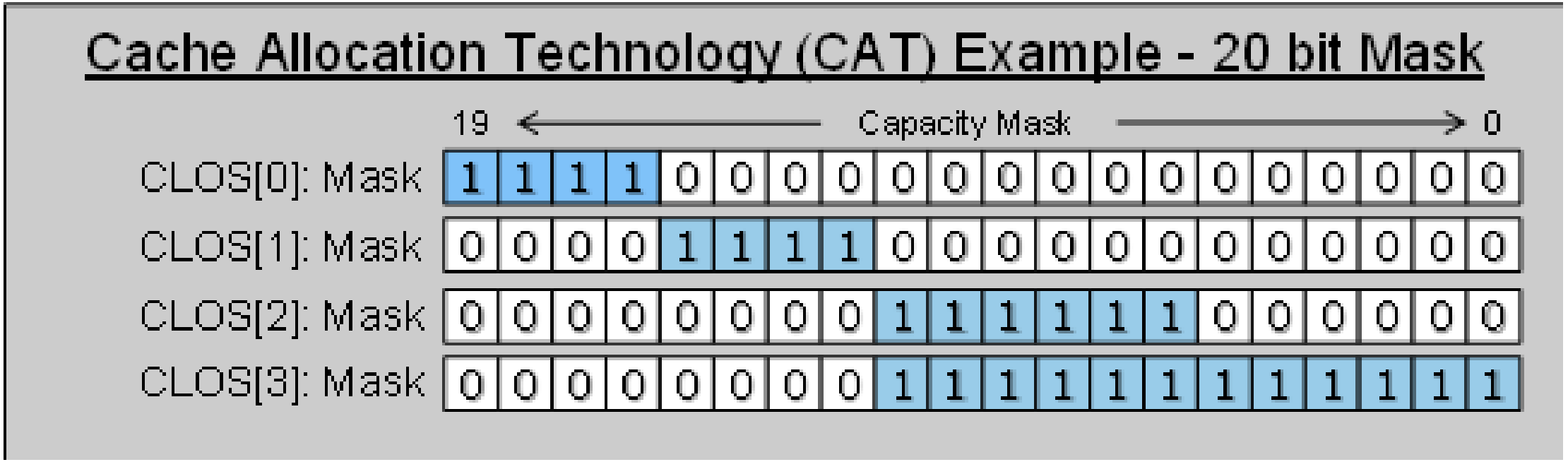
- 硬件
  - Xilinx ZCU102 FPGA开发板
- 软件
  - 基于RustSBI的启动器
  - 标签化uCore-SMP
  - 负载启动器
  - 整数排序
  - 干扰程序

硬件

- 标签化RISC-V架构部署与复现指南
- PS部分
  - 四个Arm A53核心，运行Linux，辅助RISC-V部分复位启动
  - 启动完成后不再参与RISC-V部分运行，仅通过串口收发信息
- PL部分
  - 4核（Rocket Core），100MHz，2MB L2，2GB DRAM
  - 每个核上添加一个CSR（0x9C0）记录当前核上运行程序的DSID
  - 标签系统控制平面实现为MMIO，基址为0x20000

硬件

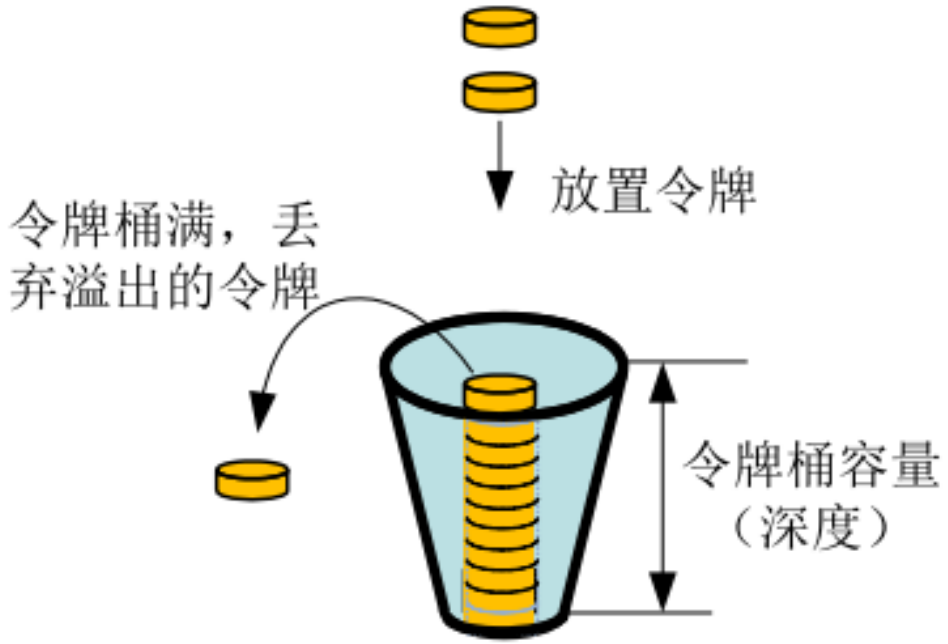
- 缓存控制：掩码

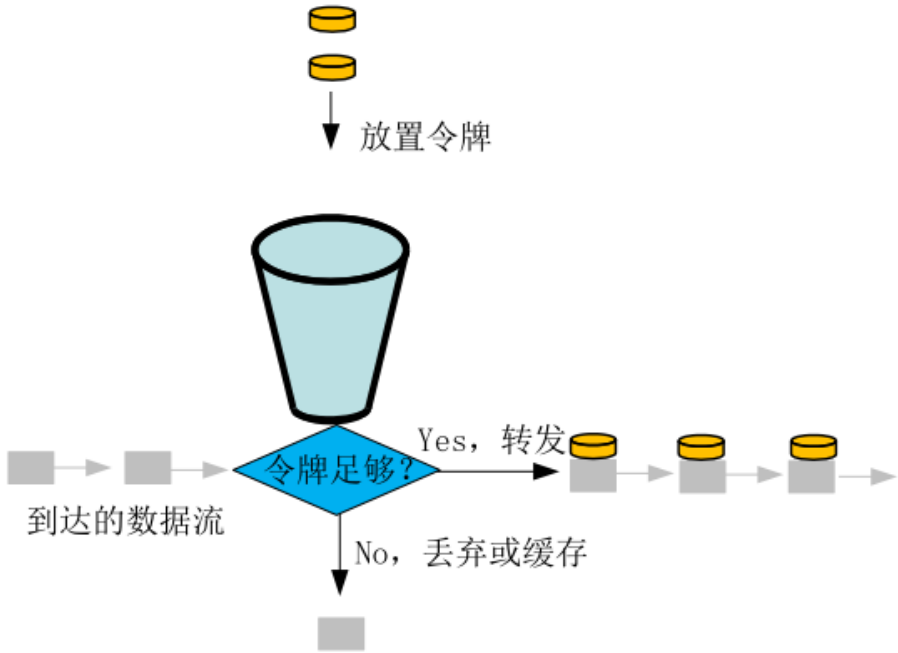


https://software.intel.com/content/www/us/en/develop/articles/cache-allocation-technology-usage-models.html

硬件

- 内存带宽控制：令牌桶算法





https://zhuanlan.zhihu.com/p/164503398

软件：启动器

- 基于RustSBI
- 内存保护初始化
  - 软复位带来的问题
  - PMP和页表
- 支持非对齐Load/Store模拟
  - 在QEMU上不存在的问题
  - 用两次对齐的Load/Store进行拼接
- Github：https://github.com/Gallium70/lrv-rust-bl

软件：标签化uCore-SMP

- 控制平面初始化
  - 恒等映射，内核态读写
- 新增3个系统调用
  - set_dsid(pid, dsid)
  - set_dsid_param(dsid, freq, size, inc, mask)
  - get_l2_traffic(dsid)
- 进程切换时更新标签
  - 内核固定为0号标签

软件：标签化uCore-SMP

- QEMU上不会遇到的问题
  - 页表项的A和D属性
  - 缓存、内存的一致性

```
[DEBUG] va=0x0000003fffffe000 npages=1 do_free=1
[TRACE 0] syscall 153 ret 11495
[TRACE 1] syscall 153 (SYS_time_ms) args:0x0000000000000000 0x0000000110000000 0x0505050505050505 0x050505050505050
5 0x0505050505050505 0x0505050505050505 0x0505050505050505
[ERROR] os/trap/trap.c:150: sepc: 0x0000000000001076, scause: 0x2, stval: 0x0000000000000000, sstatus: 0x200040120
[DEBUG 1] free_user_mem_and_pagetables free stack
panic: [TRACE 1] syscall 153 ret 11535
 Assert failed in [os/trap/trap.c:152]: "(sstatus & SSTATUS_SPP) == 0" usertrap: not from user mode[DEBUG] va=0x000
000010ffff000 npages=1 do_free=1
```

- Github：https://github.com/TianhuaTao/uCore-SMP/tree/label-riscv

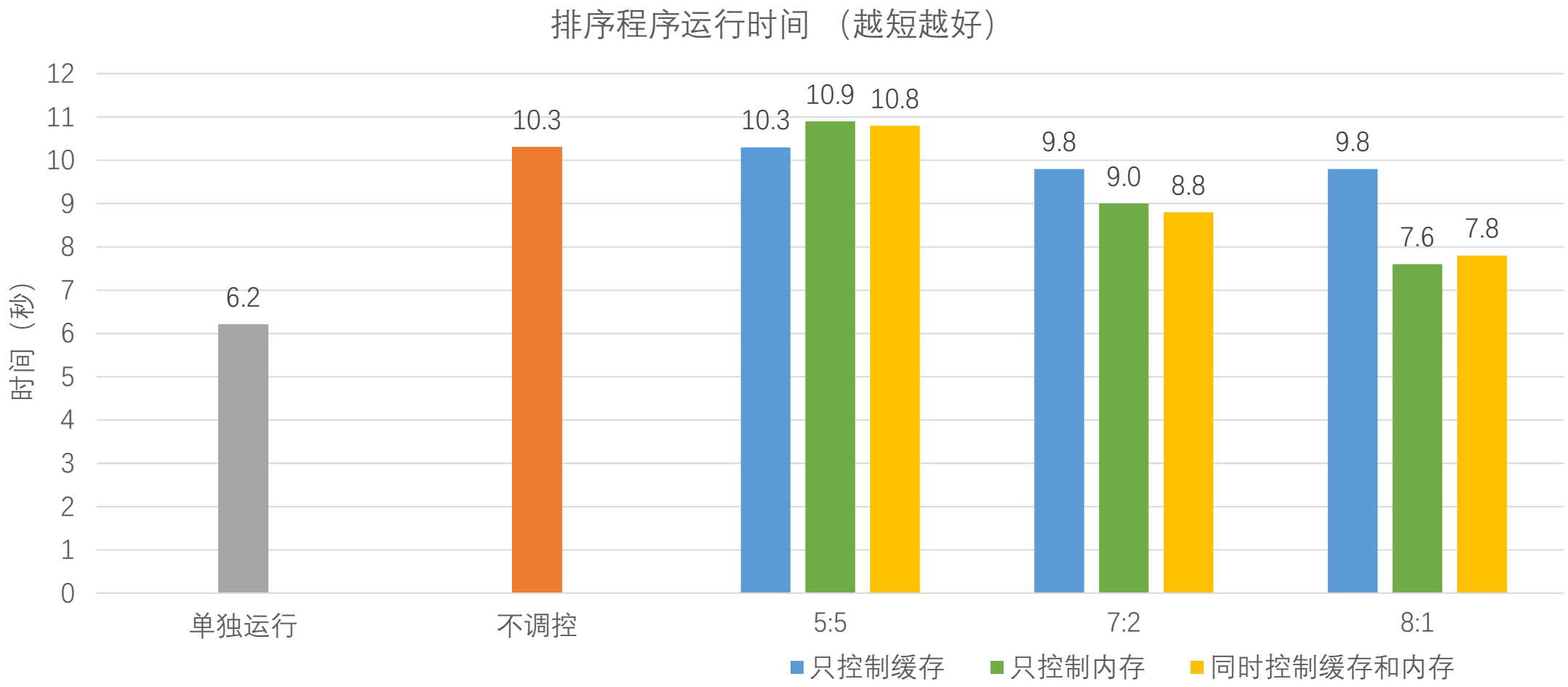
软件：负载启动器、排序和干扰程序

- 负载启动器
  - fork获取负载PID，exec启动负载，分配标签和参数
  - 监测负载使用的内存带宽（通过L1到L2流量估算）
- 整数排序
  - 500K个32位无符号整数
  - 基数排序，基为65536
  - 内存空间占用：待排数据2MB +临时空间2MB +计数数组512KB
- 干扰程序
  - 在512K长度的32位无符号整数数组上进行随机读写
  - 内存空间占用：2MB
  - 降低缓存命中率，消耗内存带宽

成果展示

- 系统内存带宽：约12MB/s
- 基准数据
  - 排序：内存带宽8MB/s，运行时间6.2s
  - 干扰程序：内存带宽8.3MB/s，运行时间6s
  - 内核和负载启动器限制到3MB/s，L2分配512K，且与负载隔离
- 四种测试场景
  - 不调控
  - 内存带宽5M：5M，缓存768K：768K
  - 内存带宽7M：2M，缓存1280K：256K
  - 内存带宽8M：1M，缓存1280K：256K

成果展示


排序程序运行时间 （越短越好）
时间（秒）
12
11
10
9
8
7
6
5
4
3
2
1
0
6.2
10.3
10.3
10.9
10.8
9.8
9.0
8.8
9.8
7.6
7.8
单独运行
不调控
5:5
7:2
8:1
只控制缓存
只控制内存
同时控制缓存和内存

后续优化方向

- 自动、闭环控制
  - 不浪费、不争抢
  - 将顶层需求描述自动转换为底层的控制参数
- 更多资源和场景
  - 图形计算、存储设备
  - 分布式场景、高实时场景
- 系统稳健性、可靠性
  - 跨核中断、Remote Fence

致谢

- 向勇、陈渝老师
  - 选题指导、框架设计、合作建议
- 张传奇
  - 标签系统从硬件配置到软件参数的保姆级支持
- 陶天骅
  - uCore-SMP、ramdisk、内核调试
- 蒋周奇、车春池
  - 启动器架构和功能建议