UINTC
用户态跨核中断控制器设计

余泰来

2021 年 12 月 11 日

用户态中断

- 中断超出一般控制流，是异步性和多主体协作能力的根本来源之一
- 应用程序也需要这些能力，但常需要内核模拟

已有工作

Intel

- 用户态跨核中断实现的 IPC 延迟是其余方式的 1/9 以下
- https://lwn.net/Articles/869140/

贺鲲鹏、尤予阳学长

- 完善实现 RISC-V N 扩展
- 已能支持 U 态外设驱动、内核分发的跨核软件中断（信号）等

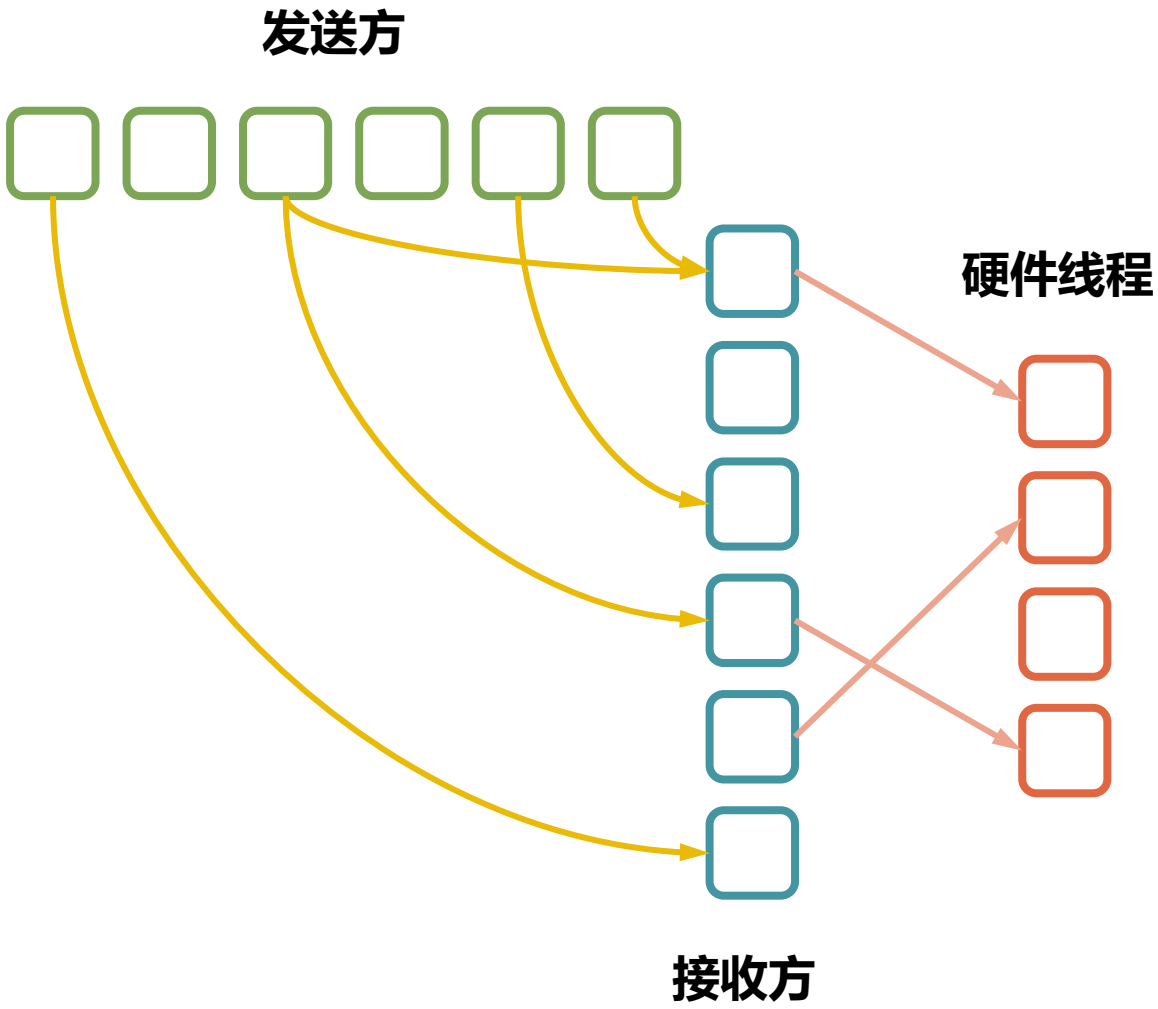
用户态跨核中断分析

特点

- 中断接收的主体是被内核“虚拟化”的应用程序，不能稳定占有硬件资源，特别是 hart
- 收发跨核中断的主体数量和 hart 数量不对等且不固定，可以单独作为发送方或接收方
- 不同发送方和接收方之间不一定有连接

解决思路

- 硬件资源需要能动态对应发送方和接收方；特别地，hart 能够动态对应中断接收方
- 将发送和接收解耦，发送方和接收方够成二分图
- 二分图的（活跃的）点和边可被内核映射到硬件上，有边连接的发送方和接收方间可以发送中断


发送方
硬件线程
接收方

x86 和 RISC-V 简要对比

Intel x86

- 对 x86 有充分主导权，增加数条指令
- senduipi 指令连续访存：
- 发送方：User Interrupt Target Table
- 接收方：User Posted-Interrupt Descriptor

- 较多中断编号，可以直接分辨中断来源

RISC-V

- 模块化，减少对不实现相关功能的处理器的影响
- 独立硬件完成中断发送——中断控制器
- 通过 MMIO 寄存器配置连接方式

- 只有 uip.USIP 一位表示中断存在，需要类似 PLIC 的 claim 机制

二分图实现

邻接矩阵

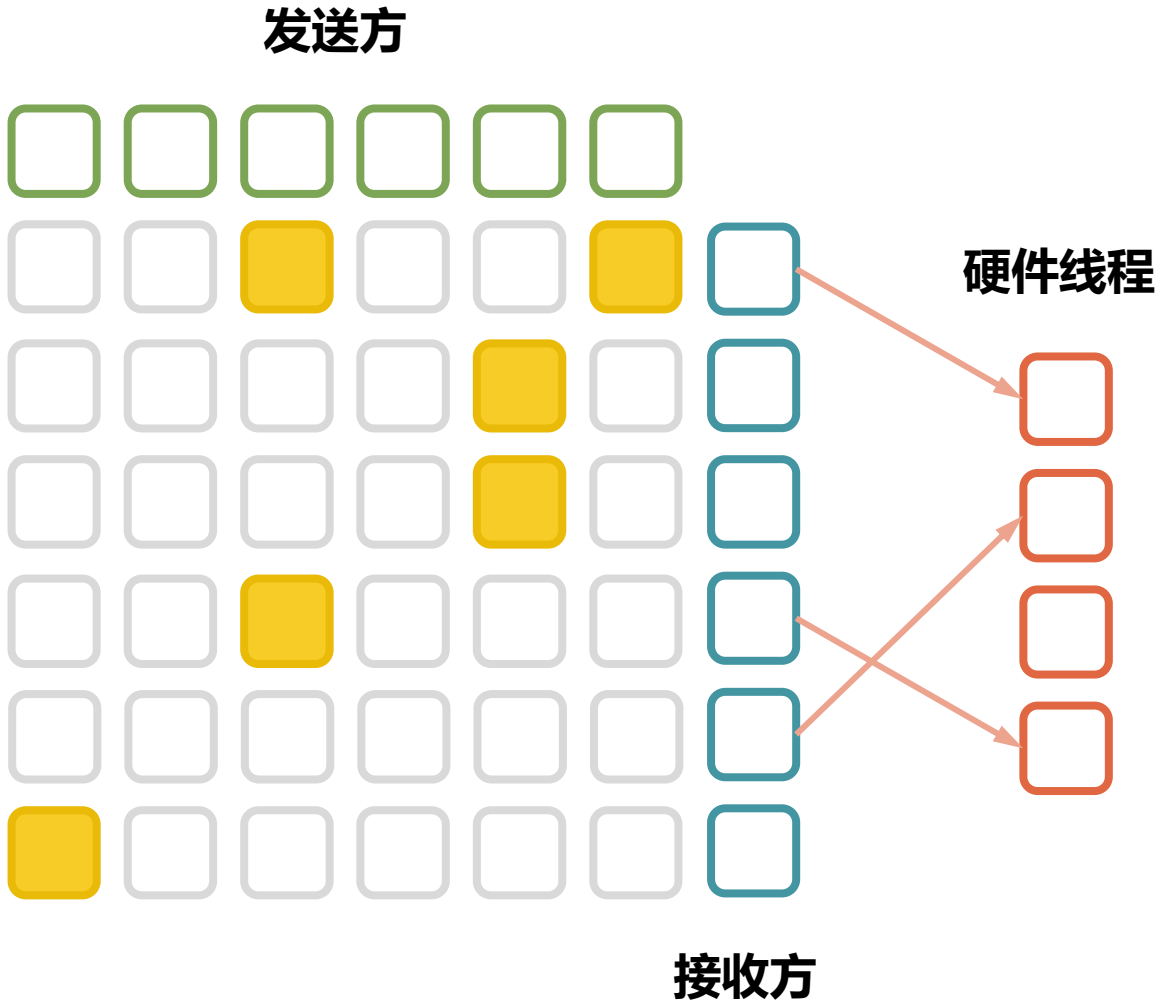
- 用 bit 矩阵表示是否有边 / 边上是否有待处理中断
- 可以支持边较多的情况
- 硬件实现较直白

邻接表

- 每个发送方有少量出边槽位，接收方有少量入边槽位，槽位中记录对方编号实现连边
- 能利用二分图稀疏性，避免浪费存储空间
- 更加动态，硬件实现较复杂

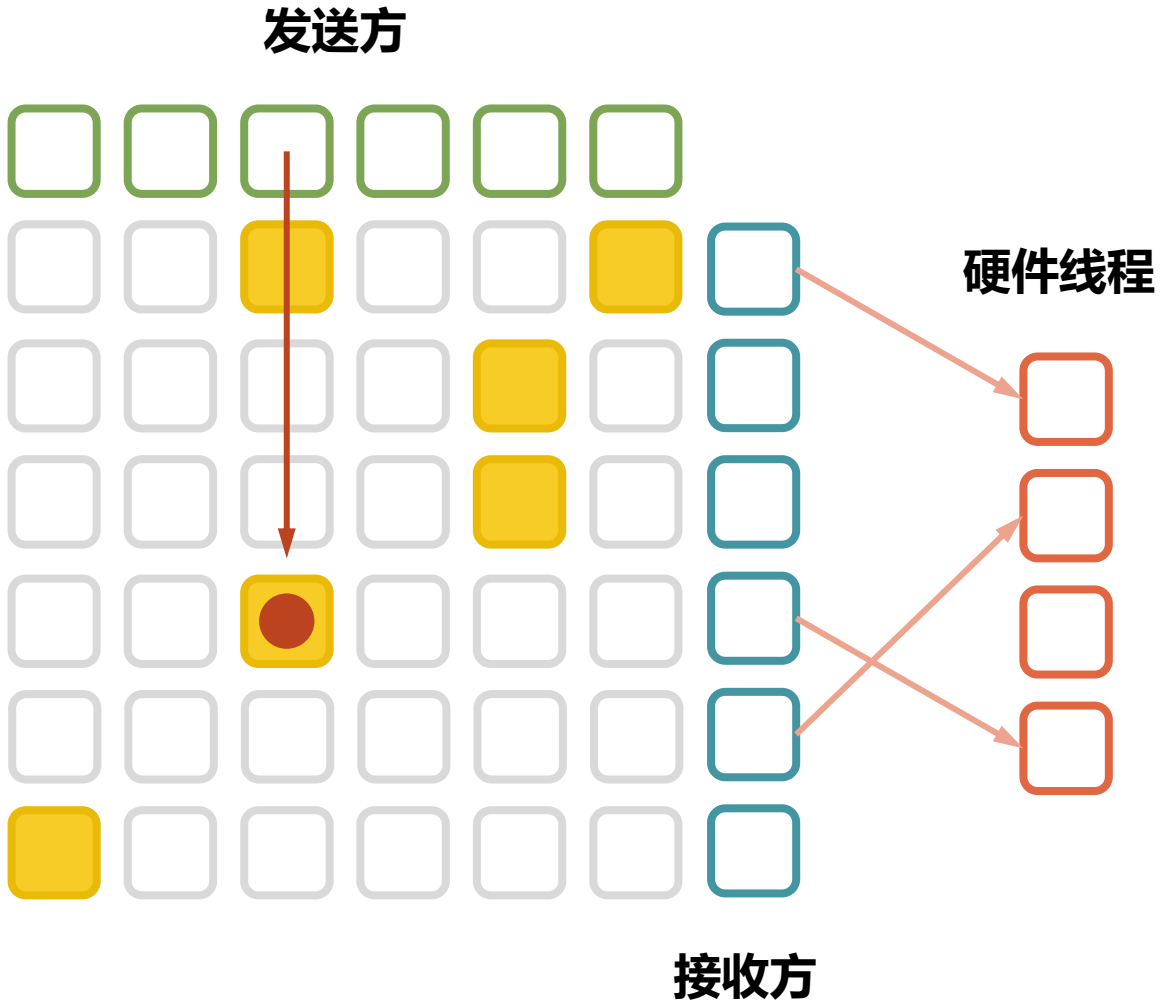
UINTC-MAT

- 发送方数 × 接收方数 的 enable / pending 矩阵
- 每个发送方 / 接收方槽位记录一软件编号 uiid，对应应用程序申请的发送 / 接收口
- 每个上下文（对应硬件线程）记录监听的接收方编号 receiver_id



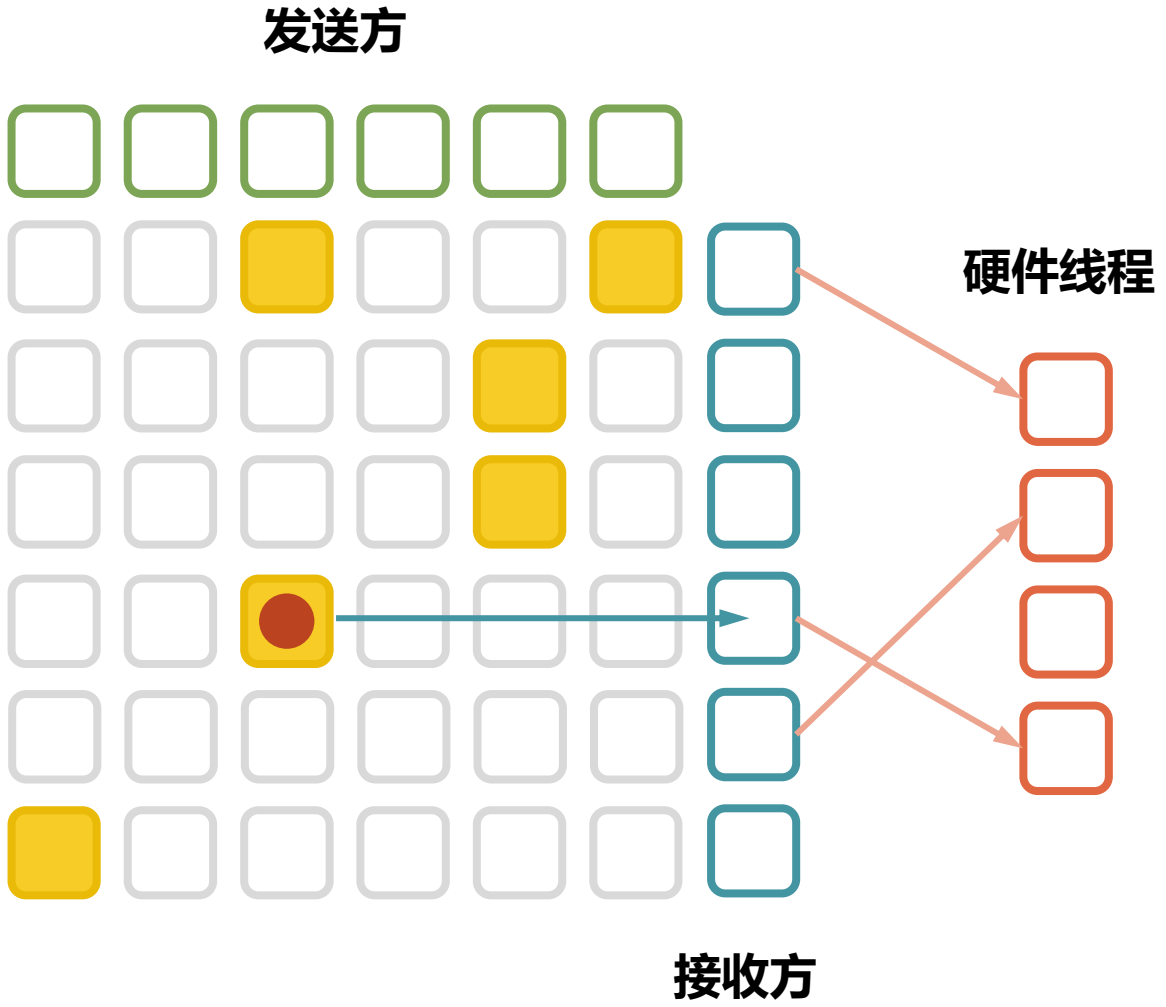
UINTC-MAT

- 每个发送方有一个 send 寄存器
- 应用程序通过向拥有的发送方对应的 send 写入合适的接收方的 uiid，向对应的接收方发送中断，如果两者有连边（enable），则置对应 pending 为 1
- 如果某个上下文的 receiver_id 为该 uiid 对应的接收方编号，该硬件线程收到中断，uip.USIP 置 1



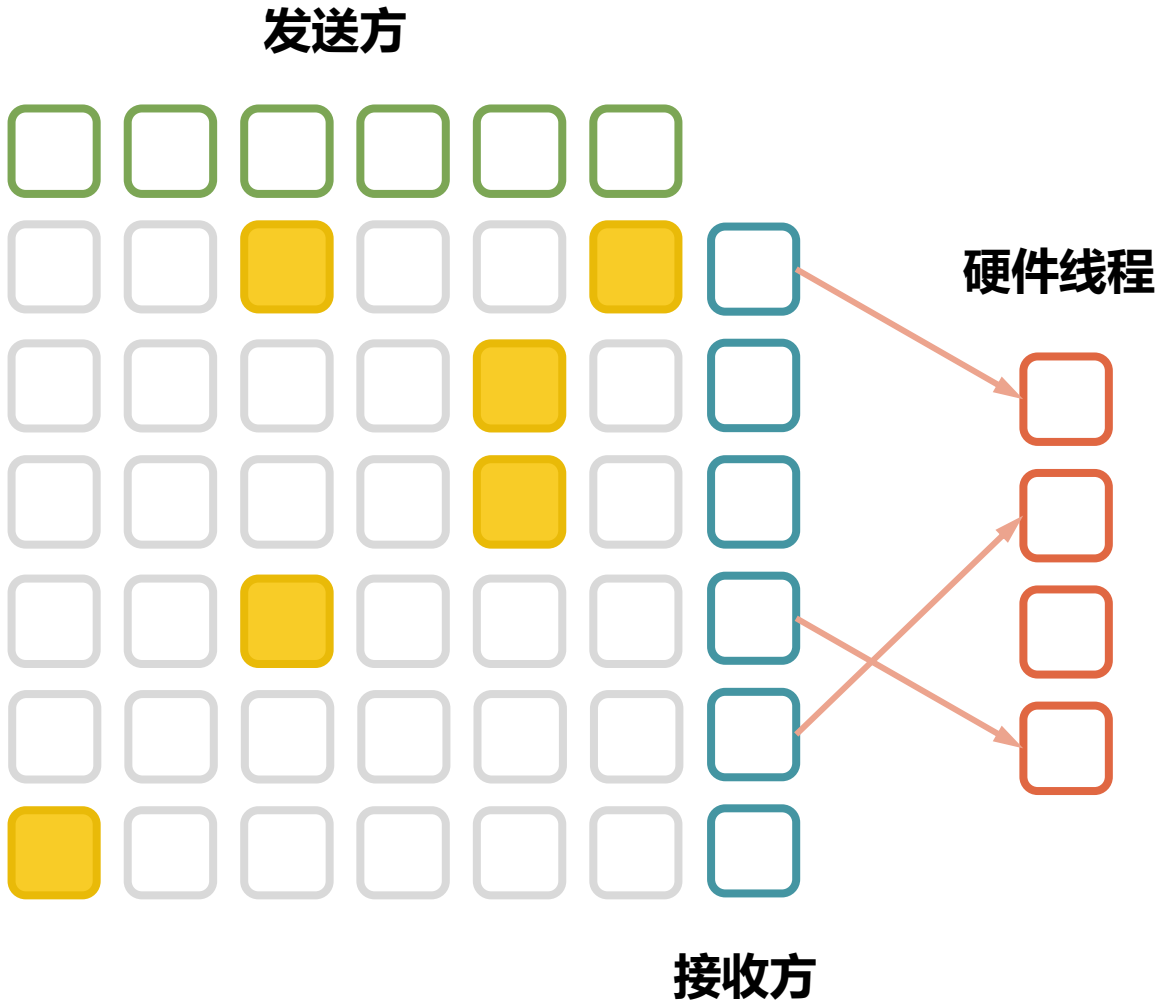
UINTC-MAT

- 每个接收方有一个 claim 寄存器。接收中断的应用程序可以读对应接收方的 claim 寄存器，读出发给此接收方的一个中断对应的发送方的 uiid
- 被读出的位置 pending 置 0



UINTC-MAT

- 操作系统可以读写 pending 和 enable，维护发送方和接收方间的连接状态
- 每个发送方和接收方可由应用程序读写的部分，单独占 4 KiB 地址空间，方便操作系统映射给各个应用程序
- 硬件提供的发送 / 接收方足够时，不在运行的应用程序仍然可以保留对应槽位的所有权
- 应用程序开始运行时，操作系统将对应硬件线程的 receiver_id 置为监听的接收方编号



操作系统接口简要设计

- UINTC 的发送方和接收方成为一种新的资源
- 通过 fork 等方式可以实现共享

- uipi_sender_ctl：创建、释放、操作发送方
- uipi_receiver_ctl：创建、释放、操作接收方
- uipi_connection_ctl：同时拥有发送方和接收方时，设置两者的连接情况

- 发送：直接在用户态写入 send
- 接收：在中断处理程序中直接在用户态读取 claim

已完成工作

- QEMU 上的模拟实现
- 在支持用户态中断的 rCore-N 基础上实现了使用 UINTC-MAT 的用户态跨核控制

待完成的工作

- FPGA 实现
- 可能会尝试邻接表实现

可能的扩展

- 此设计的动态性能够覆盖一些其他中断控制器的功能
- 通过合适连线 / 中断控制器级联，UINTC 可能可以集成高特权级或非软件来源的其他中断

谢谢大家！