产品3.0平台 - 错误 #2591

3.0 自研板 MAC TTI消息接收异常

2024-12-20 18:19 - 高 峰

状态:	挂起	**开始日期:**	2024-12-20
优先级:	高	**计划完成日期:**	
指派给:	战 弋戈	**% 完成:**	0%
类别:		**预期时间:**	0.00 小时
目标版本:		**耗时:**	0.00 小时

描述

西安魏幸幸基站3.0测试中发现

自研板卡：
（1）用SCHED_FIFO的DU版本在自研板上跑不起来，mac有很多异常。
（2）用SCHED_OTHER的DU版本在自研板，有异常但是能跑。

EVB和EVMT板卡：没有上述问题。但未确认操作系统SDK版本是否有差异

历史记录

#1 - 2024-12-24 18:25 - 高 峰

- 主题 从 MAC TTI消息接收异常 变更为 3.0 自研板 MAC TTI消息接收异常

#2 - 2025-01-06 17:34 - 战 弋戈

- 状态 从 新建 变更为 进行中

（1） 1.5B，1.5C kernel配置中，没有发现差异。
（2） phy增加打印在1.5C中基本平稳。
（3） 自研板卡增加跟evb一样的启动方式还未测试
（4）在evmb 上的测试结果还未测试，sctp 不通。

#3 - 2025-01-15 15:23 - 战 弋戈

- 文件 img_v3_02ih_89058ff2-4643-4522-a5ca-b99442c54e4g.jpg 已添加

	CPU0	CPU1	CPU2	CPU3	CPU4	CPU5	CPU6	CPU7		
11:	16553544	2331092	994910	439498	632307	137399	17425529	18195001	GICv3 27 Level	arch_timer
13:	298	0	0	0	0	0	0	0	GICv3 139 Level	ttyS0
19:	0	0	0	0	0	0	0	0	GICv3 179 Level	arm-smmu global fault, arm-smmu global fault, arm-smmu-c
20:	0	0	0	0	0	0	0	0	GICv3 181 Edge	pvt-irq
21:	2	0	0	0	0	0	0	0	GICv3 126 Level	4420000.i2c
22:	2	0	0	0	0	0	0	0	GICv3 127 Level	4428000.i2c
23:	14989369	0	0	0	0	0	0	0	GICv3 58 Level	eth0
25:	4228	0	0	0	0	0	0	0	GICv3 227 Level	eth2
36:	85706	0	0	0	0	0	0	0	GICv3 182 Level	dw-mci
37:	257257	0	0	0	0	0	0	0	GICv3 130 Level	4438000.spi
38:	7361	0	0	0	0	0	0	0	GICv3 385 Level	4d18000.spi
63:	0	0	0	0	0	0	0	0	GICv3 52 Level	dw_axi_dmac_platform
65:	0	0	0	0	0	0	0	0	GICv3 142 Edge	4400000.watchdog
66:	0	0	0	0	0	0	0	0	GICv3 143 Edge	4408000.watchdog
IPI0:	17248	4661961	14435475	1330523	2891743	3192217	10729957	12298848	Rescheduling interrupts	
IPI1:	92	3475587	81117	6829	5311	8435	883	109911	Function call interrupts	
IPI2:	0	0	0	0	0	0	0	0	CPU stop interrupts	
IPI3:	0	0	0	0	0	0	0	0	CPU stop (for crash dump) interrupts	
IPI4:	0	0	0	0	0	0	0	0	Timer broadcast interrupts	
IPI5:	0	1148304	292	317	15	266	7	527771	IRQ work interrupts	
IPI6:	0	0	0	0	0	0	0	0	CPU wake-up interrupts	
Err:	0									

在跑业务时，可以看到每个cpu的arch_timer 与 reschedule interrupt 都跑的很多，现在想法是 4，5，6核不跑中断，这个4，5，6核跑的是PHY_RECV, WORK1,WORK2 接收tti的。编码测试不绑定4，5，6核的中断没有成功。

#4 - 2025-02-06 20:10 - 战 弋戈

- 文件 20250206-200621.jpg 已添加

```
Void *FAPI_HdlMsgTransferQueue(void*tskPtr)
{
    U8 * msg_ptr = NULLP;   //这个参数应该是接受消息的地址，需要分配内存
    while(TRUE)
    {
        if(msg_transfer_receive(fapiHandle[0].value,C_PLANE,0,0,(uint8_t**)&msg_ptr) == 0)
        {
            //printf("mtTskHdlrT2kL2Fapi:msg_transfer_receive C_PLANE");
        }
        if(msg_transfer_receive(fapiHandle[0].value,U_PLANE,0,0,(uint8_t**)&msg_ptr) == 0)
        {
            //printf("mtTskHdlrT2kL2Fapi:msg_transfer_receive U_PLANE");
        }
        if(msg_transfer_receive(fapiHandle[1].value,C_PLANE,0,0,(uint8_t**)&msg_ptr) == 0)
        {
            //printf("mtTskHdlrT2kL2Fapi:msg_transfer_receive U_PLANE");
        }
        //usleep(1);
    }
    return NULLP;
} « end FAPI_HdlMsgTransferQueue »
```

根据魏幸幸的描述，去掉usleep后tti就不会看发生漂移。linux 系统中usleep的时间精度不够。

#5 - 2025-02-10 14:38 - 高 峰

usleep 精度不准的问题（400us~500us偏差），需要和思朗沟通，给出解释。
我们理解正常操作系统下，没有任务竞争的情况下，相对不会差太多的

#6 - 2025-02-20 15:29 - 战 弋戈

思朗给的反馈说，evb 跟 evmt 版本没有配置会影响usleep的精确度，linux产生的影响。

#7 - 2025-03-10 11:01 - 战 弋戈

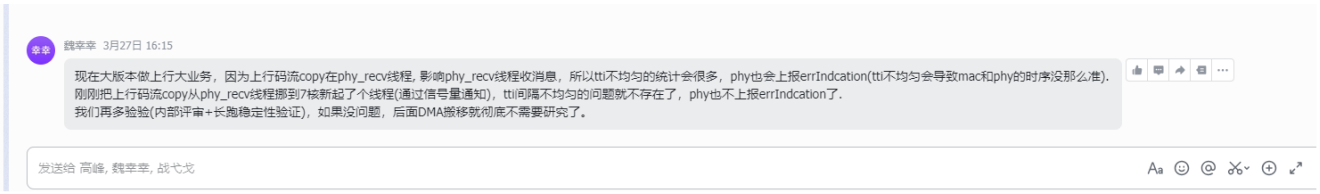
根据调度关系需要做更多测试。思朗这块没有做更多测试。

#8 - 2025-03-13 10:11 - 战 弋戈

可以从线程调度和上下文切换等方面进行测试，这些都可能引起usleep时间不准确。

#9 - 2025-04-03 14:12 - 战 弋戈

- 文件 tti幸幸修改.png 已添加



#10 - 2025-04-18 09:45 - 战 弋戈

- 状态 从 进行中 变更为 挂起

文件

img_v3_02ih_89058ff2-4643-4522-a5ca-b99442c54e4g.jpg	195 KB	2025-01-15	战 弋戈
20250206-200621.jpg	110 KB	2025-02-06	战 弋戈
tti幸幸修改.png	39.2 KB	2025-04-03	战 弋戈