IIJ’s work:
What’s new? What’s next?

2015/10/02

IIJ seil-team
Table of contents

1. What’s new in last one year?
2. MP capable network stack
3. Directions
4. Conclusion
5. Extra
1. What’s new in last one year?

• Cavium Octeon Support
• New developer: Kengo Nakahara (knakahara@)
• IRQ affinity
• MSI/MSI-X support
• PCI extended configuration support
• L2 MP stuff
• L3 MP stuff
1.1 Cavium Octeon Support

• It was first merged into OpenBSD...
• And then, it was merged into NetBSD-current by hikaru@
  – 2015/04/29
  – And then, it have been improved by matt@, martin@ and others.
• NFS problem
  – Octeon’s Ethernet controller has no TX interrupt. NFS uses m->m_ext.ext_free callback. It makes NFS slow.
  – See OCTEON_ETH_USENFS option in if_cnmac.c and nfs_vfsops.c::nfs_writerpc_extfree().
  – What’s the best way to solve this problem?
1.2 New developer

• Kengo Nakahara(\texttt{knakahara@n.o})
  – He became a developer in December 2014.
  – IRQ affinity (\texttt{intrctl(8)})
    • Merged into \texttt{--current}.
  – MSI/MSI-X
    • Merged into \texttt{current}.
  – (Ethernet)\texttt{multiqueue} for both TX and RX
    • \texttt{MI API} and \texttt{wm(4)}
    • Not merged yet.
  – Some ARM related stuff.
1.3 IRQ affinity (aka interrupt routing)

- API was defined and merged into –current with intrctl(8).
- Currently, it supports only on x86 (except Xen).
  - PowerPC/booke’s support was written by nonaka@ but not merged yet.
  - intrctl(8) has mainly 2 sub commands
    - list : show interrupts list by each CPU
    - affinity : move interrupt target to other CPU
1.4 MSI/MSI-X support

• Merged into –current
  – MI API
  – x86 MD part
    • Except Xen

• Some drivers support MSI/MSI-X
  – By knakahara@, msaitoh@ and nonaka@
  – Written but not merged yet
    • virtio(4)
    • vmxnet3(4) with multiqueue

• TODO:
  – Brush up API. Cleanup #ifdef
  – Other archs
  – Other drivers
Why is MSI-X so important?

• Because some devices support MSI-X only
  – Some server-use devices support MSI-X only
    • e.g: ixv(4)
• It’s used to make MP performance up
  – It’s not only network (Ethernet) devices
  – e.g: NVM Express and some RAID controllers
    • Someone™ should work for it 😊
    • “Multiqueue I/O in FreeBSD using LSI and NVME” by Scott Long (Saturday Track B 16:30-17:30)
Example of MSI/MSI-X and IRQ affinity (1/2)

• \textit{wm}(4)
  – It supports MSI-X for 82571 and newer devices
  – IRQ affinity is done by default.

• “\textit{intrctl affinity ...}” can be used in –current without any kernel modification.
Example of MSI/MSI-X and IRQ affinity (2/2)
1.5 PCI Extended Configuration Space support

• Why is it important?
  – To check the Advanced Error Reporting Extended Capability
    • It’s useful for debugging.
  – To control some important features
    • SR-IOV
    • M-PHY
      – Mainly used on mobile devices
    • Some power management stuff
  – PCI Exended Configuration Space is useful not only for servers but also for small devices

• We can check the detail via “pcictl pciN dump”
  – Some of them have not been decoded yet, but it decodes much more than lspci 😊
2. MP capable network stack

• Current status:
Done

• Layer 2
  – MP-safe bridge (utilizing pserialize)

• Layer 3
  – Restructure L2 XXX_output
    • pulling out rtalloc1 from XXX_output
  – Import lltable/llentry from FreeBSD
    • Toward Nexthop cache separation from the routing table

• ATF tests
ATF tests
(we added in one year)

- net/net/t_forwarding
- net/net/t_ipv6_lifetime
- net/arp/t_arp
- net/arp/t_dad
- net/icmp/t_icmp_redirect
- net/icmp/t_icmp6_redirect
- net/if/t_ifconf
- net/if/t_ifconfig
- net/if_bridge/t_bridge
- net/ndp/t_dad
- net/ndp/t_ndp
- net/route/t_flags
Pending work
(have some codes but not committed)

• MP-safe vlan (mutex)
• MP-safe bpf (mutex)
• RSTP (port from OpenBSD)
• Protect ifnet_list with pserialize
• Rump-ify netipsec for ATF tests
Future work (1/2)

• MP-safe IP forwarding
  – Nexthop cache separation from the routing table
    • Get rid of RTF_CLONING / RTF_CLONED
    • Simplify route.c
  – Coarse-grain locking on the routing table (and rtcache)
  – Protect struct ifnet and ifaddr (if required)

• MP-safe tunnels
  – gif
  – Netipsec

• MP-safe opencrypto
Future work (2/2)

• Make “options NET_MPSAFE” by default
  – We have to consider it for all network device drivers...

• IIJ has no plan to work for L4 MP
  – so please someone™
Schedule?

- Merge RX multiqueue
- Merge TX multiqueue
- Make other physical interfaces MP capable

L2 MP
- Code cleanup
- Restructure L2 xxx_output
- Next hop cache separation
- locking
- MP IP forwarding
- MP opencrypto
- MP safe virtual interfaces
- ATF test
- ATF test

L3 MP
- Make NET_MPSAFE by default
  Spring 2016(hopefully)

Need your help!

We are here now
Need discussion

• Enlarge ifnet#if_flags
• Aggregate interface packet counting
• Make if_link_state_change softint
• Nexthop cache separation
• Softint-based RX (and TX?)
• Replace the routing table
3. Directions

For embedded device

Server function

Network function

For desktop
4. Conclusion

• We merged some important functions into – current.
• Some code have not merged yet. We will merge them in future.
• Need your help.
  – Modify existing drivers
  – Network stack
  – ATF test for networking
  – Discussion
• Next target: Server functions
5: SEIL/BPV4(1/3)

- Press release (Sorry in Japanese):
SEIL/BPV4(2/3)

- Intel C2558(Rangeley)
  - qat(4): Intel Quick Assist Technology Driver
    - Written from scratch
    - Not merged into –current yet
  - /dev/random uses Intel rdrand instruction
    - Not merged into –current yet
SEIL/BPV4(3/3)

```
dmesg | grep qat
qat0 at pci:0 dev:11 function:0: Intel C2000 QuickAssist Physical Function (rev. 0x02)
qat0: sku 1 accel 1 accel_mask 0x1 ae 1 ae_mask 0x1
qat0: accel capabilities 10f<CRYPTO_KEY_AUTHENTICATION,CIPHER,CRYPTOASYMMETRIC,CRYPTO_Symmetric>
qat0: region #1 bar 0x18 CAP_GLOBAL_CTL size 0x4000 at 0x0fe80000 mapped to 0x0d092f000
qat0: region #3 bar 0x18 SSU size 0x8000 at 0x0f8e80000 mapped to 0x0d0933000
qat0: region #4 bar 0x18 AE size 0x8000 at 0x0f8e80000 mapped to 0x0d0936000
qat0: region #6 bar 0x18 EP size 0x1000 at 0x0f8e80000 mapped to 0x0d0943000
qat0: region #8 bar 0x20 PETRINGCSR size 0x40000 at 0x0xff100000 mapped to 0x0d0944000
qat0: bank0 interrupting at msix0 vec 0
qat0: bank1 interrupting at msix0 vec 1
qat0: bank2 interrupting at msix0 vec 2
qat0: bank3 interrupting at msix0 vec 3
qat0: bank4 interrupting at msix0 vec 4
qat0: bank5 interrupting at msix0 vec 5
qat0: bank6 interrupting at msix0 vec 6
qat0: bank7 interrupting at msix0 vec 7
qat0: aeccluster0 interrupting at msix0 vec 16
qat0: uof objchip_qat_nae_b0.uof at 0x600c2df0 size 0x1025c
qat0: uof at 0x600d2e10 size 0x1023c
qat0: uof cpu_type 0x00800000 min_cpu_ver 0x0000 max_cpu_ver 0x00ff
qat0: uof_image name ioc_security_noopm_bx
qat0: uof_image ae_assign 0x000000003 ctx_assign 0x00000000 cpu_type 0x00800000
qat0: uof_image max_ver 0x00000000 min_ver 0x00000000 ae_mode 0x00002018
qat0: uof_image pages 0x00000001 page regions 0x0000001
qat0: ae 0x0c5293a1e8 slice 0 page 0 assign: region 0
qat0: alloc: bank 0 ace 1 ace_mask 0x1
qat0: allocate ring 0 of bank 0 for ace10 admin_tx size 16384 16384 at vaddr 0x0d1d3000 paddr 0x49e0000
qat0: allocate ring 1 of bank 0 for ace10 admin_rx size 16384 16384 at vaddr 0x0d1d7000 paddr 0x49e4000
qat0: update intr mask for bank 0 (coalescing time 10000ns): 0x00000002
qat0: allocate ring 4 of bank 0 for cv0 smlhi_tx size 32768 32768 at vaddr 0x0d1db000 paddr 0x49e8000
qat0: allocate ring 5 of bank 0 for cv0 smlhi_rx size 32768 32768 at vaddr 0x0d1d8000 paddr 0x49f0000
qat0: update intr mask for bank 0 (coalescing time 10000ns): 0x00000002
qat0: allocate ring 8 of bank 0 for cv0 smllo_tx size 32768 32768 at vaddr 0x0d1e8000 paddr 0x49f3000
qat0: allocate ring 7 of bank 0 for cv0 smllo_rx size 32768 32768 at vaddr 0x0d1f3000 paddr 0x4a00000
qat0: update intr mask for bank 0 (coalescing time 10000ns): 0x00000000
qat0: activ writing uof ioc_security_noopm_bx
qat0: loaded firmware: Binary assembled on Oct 17 2013 at 10:02:13 Using tool version 3.0.183 . Firmware version: 1.4.0
qat0: Initialization completed
```