IIJ’s work

2014/09/26

IIJ seil-team
1. New developer
2. Yet another cvs2git
3. MP capable network stack
   1. L2
   2. MSI/MSI-X
   3. IRQ affinity
4. Future plan
1. New developer

• Hikaru Abe(hikaru@n.o)
  – He became a developer in last May.
  – ported OpenBSD’s vmx(4) to NetBSD.
  – Cavium Octeon support have not merged yet.
  • Q: When?
  • A: Ask him 😊
2. Yet another cvs2git

- We tried some tools but all of them didn’t satisfy our requirement.
- [https://github.com/IIJ-NetBSD/cvs2git](https://github.com/IIJ-NetBSD/cvs2git)
- Written by ryo@n.o
  - He aslo commited I.MX6 support yesterday.
    - (though that work is not related to IIJ)
  - This tool convert ,v to json first, so it might be easy to convert yet another VCS’s.
- It support branches
  - Not all of branches in NetBSD CVS can be syncronized (because of broken ,v?). The following branches are ok:
    - maintrunk
    - netbsd-[367]
    - rmind-smpnet

- No any document yet 😞
Collaboration space

- [https://github.com/IIJ-NetBSD](https://github.com/IIJ-NetBSD)
  - Using with ryo’s cvs2git
  - Synchronise twice a day
  - Sometime stops because of cvs2git’s debugging
  - Some of project are forked and shared with other people. e.g.:
    - L2 improvement
    - MSI/MSI-X
    - IRQ affinity
    - etc.
3. MP capable network stack

• Current status:
Layer2

- **Done**
  - Restructuring \{ether,bridge\}_input
  - MP-aware bridge (MP-safe and pktqueue)
  - Fixes of vlan(4)
  - Preliminary ATF tests for bridge
- **Future plan**
  - L3-capable bridge (like OpenBSD’s vether(4))
  - Restructuring \{ether,bridge\}_output (if possible)
  - RSTP (port from OpenBSD)
MSI/MSI-X support (1/4)

• motivation
  – for NIC multi queue

• discussing API
  – dyoung@n.o API (bus_msi(9))
    • see http://mail-index.netbsd.org/tech-kern/2014/06/06/msg017209.html
  – my API (like FreeBSD)
    • see http://mail-index.netbsd.org/tech-kern/2014/07/10/msg017336.html

• prototyping for POC
MSI/MSI-X support (2/4)

- 1st step MSI-X support for if_wm
  - separate interrupts to RX, TX and Link state

- as side effect, L2 forwarding RTT improve
  - normal interrupt: 0.579ms (10 ping average)
  - MSI-X: 0.384ms (10 ping average)
MSI/MSI-X support (3/4)

• therefore intrctl can move MSI-X to other CPUs same as normal interrupts

• determined API, implementation does not take so much time
MSI/MSI-X support (4/4)

• implementing code is here
  – with if_wm support
    • https://github.com/knakahara/netbsd-src/tree/k-nakahara-msi-msix-proto2-test-wm

• TODO
  – fallback for old PCI bridge which does not support MSI/MSI-X
  – support remapping MSI-X vectors
  – support pending bit of MSI/MSI-X
  – support special ether controller (eg 82574L)
 IRQ affinity (aka interrupt routing) (1/3)

• motivation
  – **simple** interrupts load balancing
    • this feature can distribute 4 IRQ to 4 CPUs, therefore it can reduce interrupts load to 1/4
  – for MP-safe L2 test
    • see http://mail-index.netbsd.org/tech-kern/2014/06/03/msg017190.html

• implemented intrctl(8)
  – see http://mail-index.netbsd.org/tech-kern/2014/09/12/msg017653.html
  – intrctl(8) has mainly 2 sub command
    • list : show interrupts list by each CPU
    • affinity : move interrupt target to other CPU
IRAQ affinity (aka interrupt routing) (2/3)

- example 1: intrctl command format

```
netbsd-rangeley# ./intrctl list
interrupt name CPU#00(+) CPU#01(+) CPU#02(+) CPU#03(+) CPU#04(+) CPU#05(+) CPU#06(+) CPU#07(+)
icapid 0 pin 3 0* 0 0 0 0 0 0 0
icapid 0 pin 4 0* 0 0 0 0 0 0 0
icapid 0 pin 18 0* 0 0 0 0 0 0 0
icapid 0 pin 19 1708* 0 0 0 0 0 0 0
icapid 0 pin 23 1040* 0 0 0 0 0 0 0
icapid 0 pin 22 0* 0 0 0 0 0 0 0
icapid 0 pin 21 0* 0 0 0 0 0 0 0
icapid 0 pin 20 0* 0 0 0 0 0 0 0
icapid 0 pin 9 0* 0 0 0 0 0 0 0

netbsd-rangeley# ./intrctl affinity -c 2 -i 'icapid 0 pin 23'

netbsd-rangeley# ./intrctl list
interrupt name CPU#00(+) CPU#01(+) CPU#02(+) CPU#03(+) CPU#04(+) CPU#05(+) CPU#06(+) CPU#07(+)
icapid 0 pin 3 0* 0 0 0 0 0 0 0
icapid 0 pin 4 0* 0 0 0 0 0 0 0
icapid 0 pin 18 0* 0 0 0 0 0 0 0
icapid 0 pin 19 1708* 0 0 0 0 0 0 0
icapid 0 pin 23 1040* 0 0 0 0 0 0 0
icapid 0 pin 22 0* 0 0 0 0 0 0 0
icapid 0 pin 21 0* 0 0 0 0 0 0 0
icapid 0 pin 20 0* 0 0 0 0 0 0 0
icapid 0 pin 9 0* 0 0 0 0 0 0 0
```
IRQ affinity (aka interrupt routing) (3/3)

- example 2: interrupts distribution
4. Future plan(1)

• We’d like to merge MSI/MSI-X and IRQ affinity stuff into –current.
  – to be tested by many people
  – to add MSI/MSI-X support into drivers other than wm(4)
    – to implement MD part other than x86
      • It’s required to check the validity of API
      • sparc64 support might be done by k-nakahara
• Is it time to make k-nakahara a developer?
Future plan(2)

• High priority
  – Virtual machine related improvement
    • e.g.
      – Improvements of vmx(4)
  – Add multi queue support into wm(4)
  – pseudo interface
    • ppp, pppoe
  – opencrypt, netipsec