FreeBSD Network Performance Tuning

Sucon 2004
Zurich, Switzerland

Hendrik Scholz
hscholz@raisdorf.net
http://www.wormulon.net/

Agenda

- Motivation
- Overview
- Optimization approaches
- sysctl() tuning
- Measurement
- NIC comparision
- Conclusion

Motivation

- networks are becoming faster
- time spent by CPU to handle network increases
- less network overhead -> more time for real work

end-to-end performance limited by hosts

not by network

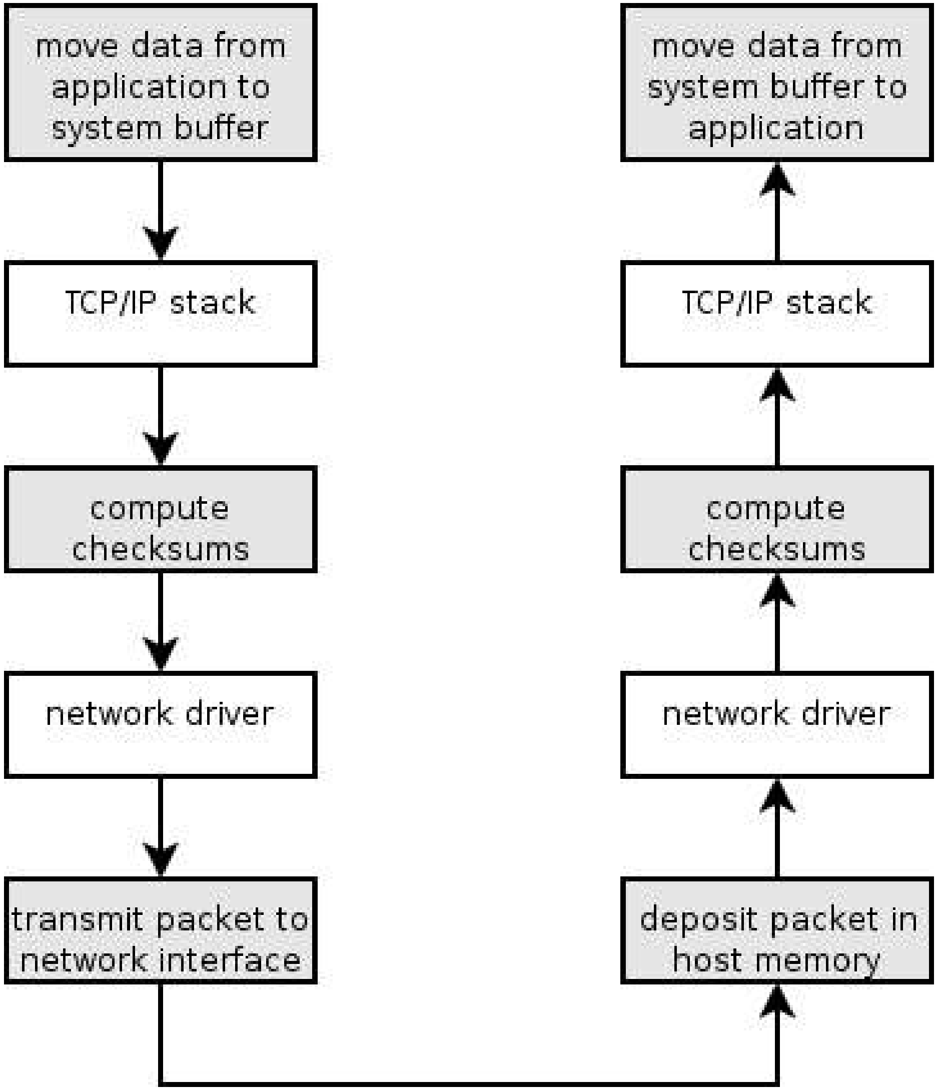
Use Cases

- ftp server
  - read data off disk and send to client
- httpd
  - send dynamic content to client
- NFS server
  - small transactions

per packet vs. per byte overhead

FreeBSD Network Performance Tuning

Overview


move data from application to system buffer
TCP/IP stack
compute checksums
network driver
transmit packet to network interface
move data from system buffer to application
TCP/IP stack
compute checksums
network driver
deposit packet in host memory

source: http://www.cs.duke.edu/ari/publications/end-systems.pdf

Maximum Transmission Unit

- Goal: better header to payload ratio
- MTU of 1500 Bytes default on Ethernet
- 9000Bytes = 'Jumbo frames'
- Jumbo frames require Gigabit equipment
- Fragmentation needed for DSL/Modem/...

```
$ ifconfig bge0 mtu 9000
```

⚠ MTU has to be consistent on LAN

Checksum Offloading 1/2

- TCP/IP/UDP require checksums
- per-byte overhead
- send packets w/o checksum to NIC
- checksum generation done in NIC
- IP checksum includes TTL thus has to be redone on each hop

⚠ invalid checksums in tcpdump
might be misleading

Checksum Offloading 2/2

- check checksum on reception in NIC
- drop invalid packets

drawbacks:

- data has to be DMAed to NIC before it can be sent/checksummed
- transmission to wire delayed

Scatter / Gather

- gather: gather header and payload from different memory addresses when sending
- scatter: store header and payload in two different memory-aligned buffers on reception
- used by sendfile() system-call
- has to be supported by driver and application

Interrupt Mitigation

- also known as Interrupt Moderation
- Aim: reduce amount of overhead for receiving packets
- swallow Interrupts and store packets in FIFO
- trigger Interrupt when FIFO full
- drawback: adds RX delay

```
[0.22] root@goanna:~ > ifconfig fxp0 link0;dmesg|tail -1
fxp0: Microcode loaded, int_delay: 1000 usec  bundle_max: 6
[0.21] root@goanna:~ >
```

Interrupt Sharing

- multiple NICs of the same type share one Interrupt
  - i.e. Adaptec 6944A, 62044, Sun QFE
  - multiple single port cards
- driver cannot determine the source of the interrupt
- has to check all cards but only has to load driver once
- use in conjunction with Interrupt Mitigation

Polling

- turns of RX interrupts from NIC
- relies on OS to service device whenever needed
- combine with IRQ sharing on QFEs
- configuration:

```
options DEVICE_POLLING

options HZ=1000
```

- not implemented: switch from polling to interrupt driven mode and back depending on load

FreeBSD Network Performance Tuning

- increase TCP sendspace to prevent blocking write()

```
$ sysctl net.inet.tcp.sendspace=65536
```

- enable RFC 1323 for larger window sizes

```
$ sysctl net.inet.tcp.rfc1323=1
```

- disable Nagle Algorithm in Samba

```
socket options = TCP_NODELAY
```

- disable/enable polling

```
$ sysctl kern.polling.enable=0
```

Measurement

- systat -if 1; systat -ip 1; systat -tcp 1
- /usr/ports/net/slurm
- sysctl net.inet
- netstat -m (sendfile buffers)

NIC comparison matrix

driver	MTU	Checksum	IRQ M.	Polling	notes
dc	1500		no	yes	
fxp	1518	(IP)|TCP|UDP	yes	yes	
ti	9018	IP|TCP|UDP	yes	no	1
bge	9018	IP|TCP|UDP	yes	no	
em	16114	IP|TCP|UDP	yes	yes	2,3
xl0	1518	IP|TCP|UDP	no	no	
re	9018	IP|TCP|UDP	no	yes	3
rl	1500		no	yes	4

1) TI_JUMBO_HDRSPLIT
2) UDP checksum limitations
3) TCP segmentation offloading
4) "redefinition of low end"

Conclusion

- Spending money on NICs makes sense
- fxp(4) and bge(4) are good and cheap
- em(4) always a good choice
- go with Gbit to benefit from additional features
- there is more do explore, i.e.
  - TCP segmentation offloading
  - SACK (RFC 2018)

Links

- Slides:

  http://www.wormulon.net/publications/sucon/
- tuning(7)
- http://www.watson.org/~robert/freebsd/netperf/
- http://www.cs.duke.edu/ari/publications/end-system.pdf