

Much improved sendfile(2) kernel implementation

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2006-09/msg00260.html>

- *From:* Andre Oppermann <andre@xxxxxxxxxxxx>
 - *Date:* Wed, 20 Sep 2006 23:59:13 +0200
-

The recent addition of TSO (TCP Segmentation Offload) has highlighted some shortcomings in our sendfile(2) kernel implementation. The current code simply loops over the file, turns each 4K page into an mbuf and sends it off. This has the effect that TSO can only generate 2 packets per send instead of up to 44 at its maximum of 64K.

I have rewritten kern_sendfile() to work in two loops, the inner which turns as many pages into mbufs as it can up to the free send socket buffer space. The outer loop then drops the whole mbuf chain into the send socket buffer, calls tcp_output() on it and then waits until 50% of the socket buffer are free again to repeat the cycle. This way tcp_output() gets the full amount of data to work with and can issue up to 64K sends for TSO to chop up in the network adapter without using any CPU cycles. Thus it gets very efficient especially with the readahead the VM and I/O system do.

Looking at the benchmarks we see some very nice improvements (95% confidence):
45% less cpu (or 1.81 times better) with new sendfile vs. old sendfile (non-TSO)
83% less cpu (or 5.7 times better) with new sendfile vs. old sendfile (TSO)

The sender is an AMD Opteron 852 (2.6GHz) with em(4) PCI-X-133 interface and the receiver is a DELL Poweredge SC1425 P-IV Xeon 3.2GHz with em(4) LOM connected back to back at 1000Base-TX full duplex.

The patch is available here:

<http://people.freebsd.org/~andre/sendfile-20060920.diff>

Any testing and heavy (code) beating and reviews welcome.

—

Andre

Here are the raw numbers (netperf at 95% confidence, +-2.5% error margin, the cpu load reported by netperf is different from the one reported by time(1), all performance references are made based on time(1) output, netperf 2.4.2 used):

- a) is old sendfile(2) kernel implementation
- b) is new sendfile(2) kernel implementation

Much improved sendfile(2) kernel implementation

- 1) time ./netperf -H192.168.2.2,4 -tTCP_STREAM -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s32K -S32K [non-TSO]
- 2) time ./netperf -H192.168.2.2,4 -tTCP_STREAM -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s32K -S32K [TSO]
- 3) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s32K -S32K [non-TSO]
- 4) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s32K -S32K [TSO]

- 5) time ./netperf -H192.168.2.2,4 -tTCP_STREAM -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s64K -S64K [non-TSO]
- 6) time ./netperf -H192.168.2.2,4 -tTCP_STREAM -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s64K -S64K [TSO]
- 7) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s64K -S64K [non-TSO]
- 8) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s64K -S64K [TSO]

- 9) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s64K -S64K -m1M [non-TSO]
- 10) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s64K -S64K -m1M [TSO]
- 11) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s64K -S64K -m2M [non-TSO]
- 12) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s64K -S64K -m2M [TSO]
- 13) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s64K -S64K -m5M [non-TSO]
- 14) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s64K -S64K -m5M [TSO]

- 15) time ./netperf -H192.168.2.2,4 -tTCP_STREAM -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s128K -S128K [non-TSO]
- 16) time ./netperf -H192.168.2.2,4 -tTCP_STREAM -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s128K -S128K [TSO]
- 17) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s128K -S128K [non-TSO]
- 18) time ./netperf -H192.168.2.2,4 -tTCP_SENDFILE -C -c -F 6.2-BETA1-i386-disc1.iso
-- -s128K -S128K [TSO]

Recv	Send	Send	Utilization	Service	Demand						
Socket	Socket	Message	Elapsed	Send	Recv	Send	Recv				
Size	Size	Size	Time	Throughput	local	remote	local	remote			
bytes	bytes	bytes	secs.	10^6bits/s	%	C	%	C	us/KB	us/KB	

1) 32768 32768 32768 10.00 921.16 28.27 31.88 2.514 2.835
0.000u 1.703s 0:10.00 17.0% 94+5091k 0+0io 0pf+0w

2) 32768 32768 32768 10.00 897.91 23.83 38.65 2.175 3.526
0.000u 1.310s 0:10.02 13.0% 91+4925k 0+0io 0pf+0w

Much improved sendfile(2) kernel implementation

Much improved sendfile(2) kernel implementation

3) 32768 32768 32768 10.00 767.31 20.98 29.92 2.240 3.195
0.013u 0.969s 0:10.00 9.7% 109+5855k 0+0io 1pf+0w

4a) 32768 32768 32768 10.00 911.66 15.71 33.61 1.412 3.020
0.000u 0.651s 0:10.00 6.5% 93+4993k 0+0io 0pf+0w

4b) 32768 32768 32768 10.00 759.08 11.13 30.70 1.201 3.313
0.007u 0.266s 0:10.00 2.6% 108+5796k 0+0io 0pf+0w

5) 65536 65536 65536 10.00 941.59 29.17 31.73 2.538 2.760
0.000u 1.759s 0:10.01 17.4% 93+5012k 3+0io 0pf+0w

6) 65536 65536 65536 10.00 921.97 26.47 38.80 2.352 3.447
0.000u 1.401s 0:10.00 14.0% 97+5216k 0+0io 0pf+0w

7a) 65536 65536 65536 10.00 941.59 23.08 33.23 2.008 2.891
0.000u 1.178s 0:10.00 11.7% 92+4986k 0+0io 0pf+0w

7b) 65536 65536 65536 10.00 940.76 23.68 31.43 2.062 2.737
0.000u 1.202s 0:10.00 12.0% 90+4862k 0+0io 0pf+0w

8a) 65536 65536 65536 10.00 936.75 16.62 33.08 1.453 2.893
0.000u 0.611s 0:10.00 6.1% 99+5320k 0+0io 0pf+0w

8b) 65536 65536 65536 10.00 938.99 12.11 33.08 1.056 2.886
0.006u 0.245s 0:10.00 2.4% 117+6279k 0+0io 0pf+0w

9a) 65536 65536 1048576 10.00 941.59 23.61 32.53 2.054 2.830
0.000u 1.147s 0:10.00 11.4% 97+5253k 0+0io 0pf+0w

9b) 65536 65536 1048576 10.00 940.43 20.68 32.63 1.801 2.843
0.000u 1.149s 0:10.00 11.4% 93+5016k 0+0io 0pf+0w

10a) 65536 65536 1048576 10.00 931.96 16.47 35.31 1.447 3.104
0.000u 0.631s 0:10.00 6.3% 91+4906k 23+0io 0pf+0w

10b) 65536 65536 1048576 10.00 929.90 11.58 34.94 1.020 3.078
0.000u 0.201s 0:10.00 2.0% 126+6762k 0+0io 0pf+0w

11a) 65536 65536 2097152 10.00 941.59 23.29 32.26 2.026 2.806
0.000u 1.153s 0:10.00 11.5% 95+5107k 0+0io 0pf+0w

11b) 65536 65536 2097152 10.00 940.38 21.88 31.86 1.906 2.775
0.000u 1.157s 0:10.00 11.5% 94+5073k 0+0io 0pf+0w

12a) 65536 65536 2097152 10.00 936.75 16.92 33.31 1.479 2.913
0.000u 0.651s 0:10.00 6.5% 100+5409k 0+0io 0pf+0w

12b) 65536 65536 2097152 10.00 935.48 10.97 32.03 0.961 2.805
0.000u 0.201s 0:10.00 2.0% 97+5216k 20+0io 0pf+0w

Much improved sendfile(2) kernel implementation

Much improved sendfile(2) kernel implementation

13a) 65536 65536 5242880 10.00 941.58 23.68 31.13 2.061 2.708
0.000u 1.168s 0:10.00 11.6% 101+5462k 0+0io 0pf+0w

13b) 65536 65536 5242880 10.00 940.22 20.68 33.46 1.802 2.915
0.000u 1.163s 0:10.00 11.6% 91+4896k 0+0io 0pf+0w

14a) 65536 65536 5242880 10.00 923.40 17.44 33.66 1.548 2.986
0.000u 0.656s 0:10.00 6.5% 103+5528k 84+0io 0pf+0w

14b) 65536 65536 5242880 10.00 928.56 10.90 33.23 0.962 2.932
0.000u 0.206s 0:10.00 2.0% 115+6182k 64+0io 0pf+0w

15) 131072 131072 131072 10.00 941.62 33.98 31.95 2.957 2.780
0.000u 2.098s 0:10.00 20.9% 95+5102k 1+0io 0pf+0w

16) 131072 131072 131072 10.00 922.41 28.42 39.25 2.524 3.486
0.007u 1.646s 0:10.00 16.4% 91+4924k 0+0io 0pf+0w

17) 131072 131072 131072 10.00 941.61 21.88 32.68 1.904 2.843
0.000u 1.204s 0:10.00 12.0% 96+5152k 0+0io 0pf+0w

18) [the em(4) interface wedged in TSO]

freebsd-net@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-net>

To unsubscribe, send any mail to "freebsd-net-unsubscribe@xxxxxxxxxxx"