ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Study of Linux Kernel Parameters to Enhance TCP Performance

[1]Sudhakar Mishra, [2]Shefali Sonavane, [3]Anil Gupta, [4]Sweta Anmulwar, [5]Tarun Batra
[1,2]Walchand College of Engineering, Sangli, Maharashtra, India
[3,4,5]Center for Development in Advanced Computing Pune, Maharashtra, India

Abstract
Ethernet has grown from 10 Mbps to 10 Gbps surpassed the growth of microprocessor performance in mainstream servers and computers. Kernel parameter responsible for throughput comes with per-determined value. Network throughput can be increased if these values are assigned to the best possible values. Study of these parameters called as TCP tuning which adheres to find out the value for these parameters to achieve optimum performance. This paper describes parameters and their tuning through which system performance can be increased. Experiments carried out on 10 Gbps link with these setting to check the throughput performance gain.

Keywors
Traffic Generation, Internet, TCP

I. Introduction

With the advent and deployment of Gigabit links, the capacity grows by orders of magnitude. Such high-capacity links have been used to transfer huge volumes of data [1]. During the network evaluation, each card presents a good number of features which increases the performance of the network, but when not properly configured, can harm the overall system and network performance.

Linux on the other side is an operating system which runs from single machine to supercomputers and its default configuration is not set for 10 Gbps network link which possibly limits the available throughput. Default TCP parameters in most Linux distributions are conservative, and are tuned to handle 100 Mbps or 1 Gbps port speed which result in TCP buffer sizes that are too small for 10 Gbps networks. Modifying these values can lead to significant performance gain in a 10 Gbps link [4].

This paper describes few important parameters which need to be tuned to get the optimum performance. The rest of paper is organized as follows: Section II gives the overview of tuning parameters and how they affect the network performance. Section III gives the necessary information to tune the Linux kernel with command and parameters. Section IV, presents some experimental data which show the performance gain when these parameters are tuned. Finally, Section V concludes with future scope announcement.

II. Terminology and Parameters

This section covers the theory of parameters used for tuning and elaborate on how these parameters are related to performance.

A. Throughput

Throughput is the parameter which means as data transfer rate or digital bandwidth consumption. It is amount of work that a computer can perform in given period of time. The work can be measured in terms of the amount of data processed or transferred from one location to another by a computer, computer network or computer component. It's specified in bit per second (bps). Throughput can be high or low depending upon network infrastructure. Optimum throughput is required for good performance.

B. TCP Socket Buffers

TCP Socket buffer is a buffer that kernel allocates to hold the data before read/write into the given socket during the time between it arrives over the network and when it is read/write by the program that owns this socket. Receiver buffer holds incoming packets till program reads it. When this buffer gets full, receiver notifies sender to slow down. When receiver is idle, sender fill buffer and send data when receiver signals. Socket buffer plays very important role in determining the optimum throughput of the network. If buffers are small in size, packet loss may increase but larger buffer come in terms of expensive memory. Both sender and receiver buffer should be tuned to find an optimum value where maximum through put can be achieved.

C. Jumbo Frames

A jumbo frame is an Ethernet frame with a payload greater than the standard maximum transmission unit (MTU) of 1,500 bytes. Jumbo frames are used on local area networks that support at least 1 Gbps link speed and can be as large as 9,000 bytes [RFC 4638]. Use of Jumbo frames has two advantages: One is that for a given data throughput, the packet rate is less for jumbo frames as compared to standard frames, requiring less CPU for packet processing. The other advantage is that recovery in data throughput after a loss event is proportional to the maximum segment size, so with jumbo frames you get six time faster recovery rates from loss events [2]. But, larger frames consume more Ethernet link transmission time, causing greater delays for those packets that follow and thus increasing lag time and latency. This could have a negative consequence for applications that require low latency and consists of smaller packet sizes such as Voice over IP or Inter-Process Communication (IPC) [3]. A tradeoff is needed as per the application.

D. Transmission Queue Size

The transmission queue is the buffer, which holds packets that is scheduled to be sent to the Ethernet. Tuning queue size is very important to avoid the packet drop due to unavailability of memory space or fast sender. Default size of this queue is 1000 frame count. Actual size depends upon the characteristics of the network. Small value is recommended for slower devices which have high latencies like modem links and ISDN. High value is recommended for server connected over the high-speed Internet connections that perform large data transfers [5].

E. Other Parameters

Linux kernel has enumerable parameters which determine the effective throughput of the link. Table 1 presents the list of such parameters with required action.

Table 1. TCP Tuning Parameters

Parameter	Action
Bandwidth Delay Product(BDP)	TCP Buffer should be above BDP
Interrupt Affinity	Assign dedicated CPU for Interrupt handling
TCP Tx & Rx Checksum Offload	Enable
Scatter & Gather	Enable
TCP Segmentation Offload	Enable
Large Receive Offload	Enable
TCP Window Scaling	Enable
TCP SACK	Disable
MTU	Optimum Size
Transmission Queue	Optimum Size
Nagle Algorithm	ON
TCP Timestamps	Disable

TCP Tuning in Linux Kernel

In prior Linux Kernel, it was not easy to change kernel parameters. You have to recode and compile the kernel in order to tune parameters. But earlier version provides modification of these parameters using system calls [4]. The values can be set using sysctl command. Use of sysctl is usually easy as it contain /etc/sysctl.conf or any other chosen config script, which allows keeping the setting even after the machine restart. MTU and txqueuelen as modified using ifconfig command. These parameters should be restored after experiment.

```
# sysctl net.ipv4.parameter_name=Value

sysctl net.ipv4.tcp_window_scaling = 1
sysctl net.core.rmem_max = 16777216
sysctl net.ipv4.tcp_rmem= "4096 87380 16777216"
sysctl net.ipv4.tcp_wmem = "4096 16384 16777216"
```

rmem_max is the maximum TCP buffer size possible. Three values in subsequent command corresponds to minimum/average/maximum values of buffer.

```
# sudo ifconfig ethX mtu value
  sudo ifconfig eth2 mtu 9000
# sudo ifconfig ethX txqueuelen value
  sudo ifconfig eth2 txqueuelen 2000
```

III. Result and Analysis

The laboratory set up comprises of two computers (servers) with Linux (Centos 6.2, Kernel Version 2.6.32–220) operating system connected using cross over cable using TCP/IP protocol (IPv4) as illustrated in Fig. 1. The servers have Intel (R) Xeon (R) CPU with 2.93 GHz CPU and 64GB of memory using Network Card Mellanox Technologies MT25418 [ConnectX VPI PCIe 2.0 2.5GT/s - IB DDR / 10GigE. The jumbo payload (> 1500Byte s) were sent over 10Gbps Ethernet link to achieve the maximum throughput.'C' language application program UTGen is written to generate the traffic among the peers.

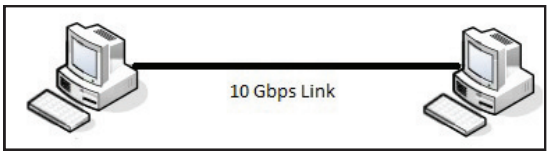

Fig. 1: Laboratory Setup

A. TCP Buffer Tuning

Figure 2 and Fig. 3 shows the effect of varying TCP buffer. Throughput is less with small packet sizes. As packet size increases, throughput increases. Receiver Buffer variation.

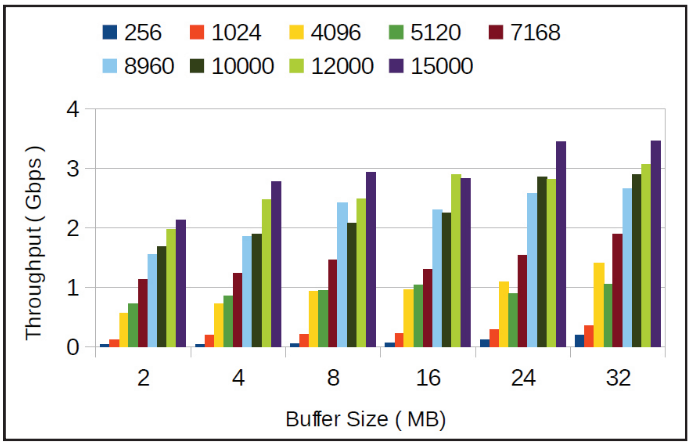

Fig. 2: Receiver Buffer Tuning

has radical effect on throughput. Maximum throughput is achieved at receiver buffer tuning is 3.55 Gbps with 15000 data buffer on 24 MB receiver buffer. Sender buffer effect is comparatively less than receiver. Maximum throughput achieved at sender buffer tuning is 3.06 Gbps at 15000 on 24 MB sender buffer. Throughput is more dependent upon the the receiver buffer as sender are usually fast. Receiver requires best possible values for buffer in order to holds incoming packet till it reads them.

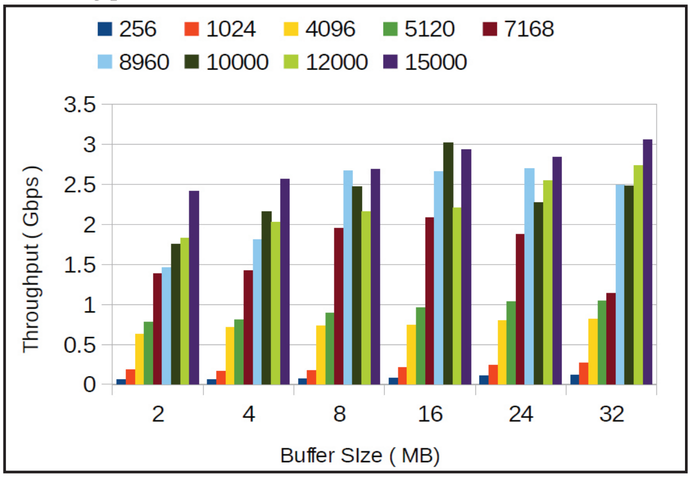

Fig. 3: Sender Buffer Tuning

B. Jumbo Frame

Fig. 4 show the jumbo frame effect on throughput. From graph inferred that Jumbo frame enhances the throughput for small packet sizes, rather than large packet sizes. It reduces the packet overhead of small packets. Maximum throughput is 3.45 Gbps at 15000 Data Buffer on 32 MB Buffer memory. Our experimental data show that Jumbo frame contribute around 10% to 30 % increase in throughput. Enhancement is more observed for packet size less than 9000 bytes, after which enhancement is insignificant.

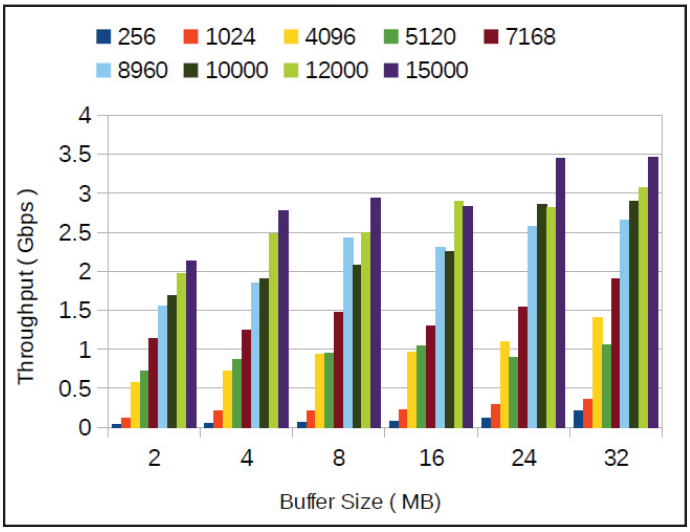

Fig. 4: Jumbo Frame Effect

C. Transmision Queue Tuning

Fig. 4 and Fig. 5 show the transmission queue effect on throughput.

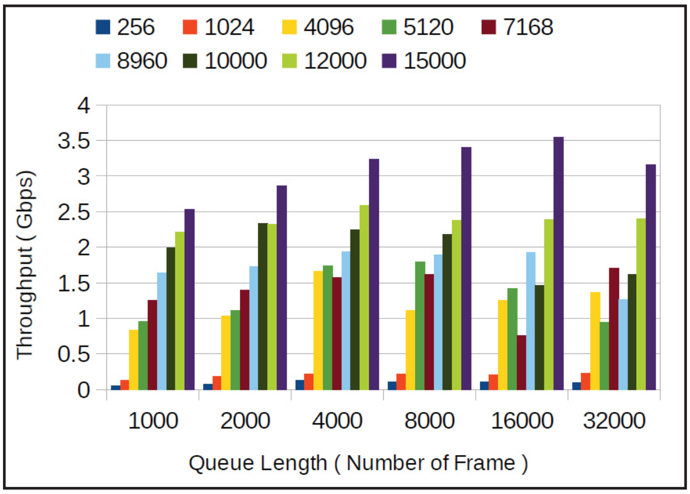

Fig. 5: Receiver Queue Tuning

As Transmission queue size increases, throughput increases. Throughput increases radically till 8000, beyond that increase is insignificant. Optimum throughput is 3.4 Gbps at 8000 queue size on sender side and 2.65 Gbps on receiver side.

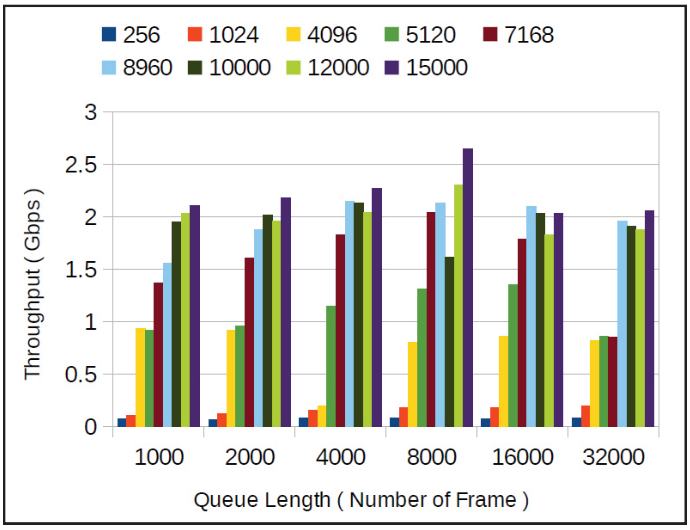

Fig. 6: Sender Queue Tuning

V. Conclusion

Network performance is very critical component of system. Resources are needed to be fully utilized in order to have maximum return on hardware investment. By applying, simple tuning mechanism performance increases dramatically. It's proved by experiments that TCP Tuning can enhance the throughput. But it requires deep analysis and study of parameters. There is no single set of rules that work in all situations. Finding suitable parameter value set for optimum throughput is a great challenge. This experimental study achieved throughput at 3.66 Gbps via TCP tuning around 3 times faster against normal condition. It's also found that receiver buffer size tuning contribute much more than sender side tuning.

VI. Future Work

Future work can incorporate multi stream data flow using multi-threading to increase the throughput .TCP behavior on IPv6 protocol can be checked and compared against IPv4.

References

[1] DataTAG project, [Online] Available: http://www.datatagorg
[2] MTU size hanging, [Online] Available: https://fasterdata.es.net/networktuning/mtu-issues/
[3] Robert Winter, Rich Hernandez, Gaurav Chawla, "Ethernet Jumb rames", Published at [Online] Available: http://www.Ethernet alliance.org, Nov, 2009.
[4] A White Paper on,"10GbE Network Performance Tuning", by Oracle at [Online] Available: http://www.oracle.com, May 2013.
[5] Transmission Queue Tuning [Online] Available: http://www.cyberciti.biz/faq/gentoo-centos-rhel-debianfedora-increasing-txqueuelen/