A Study on TCP Buffer Management Algorithm for Improvement of Network Performance in Grid Environment

Yonghwan Jeong[1], Minki Noh[2], Hyewon K. Lee[1], and Youngsong Mun[1]

[1] School of Computing, Soongsil University
1-1, Sando 5Dong, Dongjak-Gu, Seoul, 156-743, Korea (South)
{paul7931,kerenlee}@sunny.ssu.ac.kr, mun@computing.ssu.ac.kr
[2] Korea Institute of Science and Technology Information (KISTI)
Eoeun-dong 52, Yuseong-gu, Daejeon city, Korea (South)
mknoh@kisti.re.kr

Abstract. The Grid is the environment that connects high performance computing resource, which are scattered geographically as related network. The Grid, which started in the mid of 1990, has studied and laid across in BT, NT, and ET fields. The Grid applications are developed to act as global infrastructure, especially, linked as high performance network. Nevertheless, the Grid network envionment are consists of high speed researches, it uses network management method of former old internet envionment, and it cannot take the full advantage of high performance network. This research suggests TCP buffer control mechanism that is more appropriate in high performance network for better performance in The Grid network. In addition, controlled method analyzes the network performance using the Globus Toolkit 3.0, which is the most recent Grid middleware.

1 Introduction

The Grid is the environment that connects high performance computing resource, which are scattered geographically as related network. Grid, which started in the mid of 1990, has studied and laid across in BT, NT, and ET fields. The Grid computing has concepts that it makes geographycally distributed and unused high perfomance computing resourcese into available things. Its resource sharing and cooperation is accomplished by Grid network. American Abilene and vBNS, European TEN-155, SINET of Japan and KREONet of Korea achieve Grid network's functions. These networks affect high performance network's form.

In order to guarantee the network QoS of Grid application in high performance network, Grid environment provides GARA (General Purpose Architecture for Reservation). GARA provide uniform API to various types of resources to network QoS. Also, GARA adopts Differentiated Service (DiffServ) infrastructure in IETF. DiffServ guarantees Grid application's end-to-end network QoS by establishing ToS field in IP header. Nevertheless, the Grid network envionment are consists of high speed researches, Which uses network management method of former (old) internet envionment, and it cannot take the full advantage of high performance network.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3038, pp. 281–288, 2004.

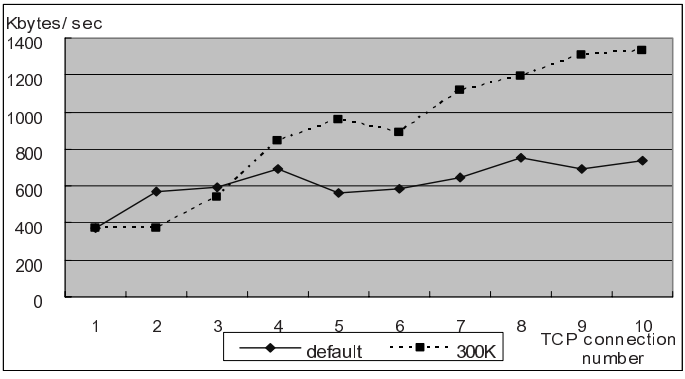

Fig. 1. TCP transmission rate variation according to number of connection

Fig. 1. shows the data transmission amount by the increasing number of TCP connection in case of configuring TCP buffer size to default size, 64 Kbytes or default size, 300Kbytes. Each case is represented as a solid line and dotted line respectively. According to Fig. 1., TCP buffer size is necessary to be modified to adopt the state of network in the high performance network such as Grid environment. As a result the data transmission rates can be improved.

This research suggests that TCP buffer control mechanism is more appropriate in high performance network for better Grid network performance. In addition, controlled method analyzes the network performance using the Globus Toolkit 3.0, which is the most recent Grid middleware.

2 Related Works

2.1 GridFTP

In Grid environment, access to distributed data is typically as important as access to distributed computational resources. Distributed scientific and engineering applications require two factors. The first factor is to transfer of large amounts of data, such as in terabytes or petabytes length data between storages systems. And the second factor is to access to large amounts of data, such as in gigabytes or terabytes data by many geographically distributed applications and users for analysis and visualizations. The lack of standard protocols for transfer and access of data in Grid has led to a fragmented Grid storage community. Users who wish to access different storage systems are forced to use multiple APIs and protocols. The performance of data transmission between these different storage systems shows a drop in efficiency; drop off in efficiency.

GridFTP, which has functionality of command data transfer and access protocol, provides secure and efficient data transmission in Grid environment. This protocol, which extended the standard FTP protocol, provides a superset of the features offered by the various Grid storage systems currently in use. The GridFTP protocol includes following features:

- Grid Security Infrastructure (GSI) and Kerberos support
- Third-party control of data transfer
- Parallel data transfer
- Support for reliable and restartable data transfer
- Integrated instrumentation

2.2 TCP Buffer Tuning

Buffer Tuning Overview. In order to decide how many packets can be sent, TCP uses "cwnd(congestion window)" parameter. The larger window size is, it is more throughputs at once. Both TCP slow start mechanism and congestion avoidance algorithm decide congestion window size. The maximum window size has relation with the buffer size of each socket. The buffer size of each socket is assigned by Kernel and has default value; this value can be modified by application before socket establishment. The application can be one of the programs using system library call. The Kernel forcese to use maximum buffer size. The buffer size can be modified by both sender and receiver.

In order to get maximum throughput, it is important to use adequate TCP sender/receiver buffer size for link. If buffer size is too small, the TCP congestion window will not open enough. On the contrary, if buffer size is too large, the TCP congestion window will close, and receiver's buffer will be overflow. The same results will be happen if sender is faster than receiver. However, whether sender's window is too large size doesn't affair in the case of enough memory. (1) means adequate buffer size.

$$\text{Buffer size} = 2 * \text{bandwidth} * \text{delay} \tag{1}$$

$$\text{Buffer size} = \text{bandwidth} * \text{RTT} \tag{2}$$

"ping" application is used to get delay value, and "pipechar" or "pchar" is used to get the last link bandwidth. (2) is identical to the (1) because RTT (round trip time) is obtained by "ping" application.

If ping time is 50ms on end-to-end network, which is composed of 100BT and OC3 (155Mbps), adequate TCP buffer size is 0.05sec * (100Mbits / 8bits) = 625 Kbytes for this connection.

There are two things to notice. They are default TCP sender/receiver buffer size, and maximum TCP sender/receiver buffer size. For the most UNIX OS, default maximum TCP buffer size is 256KB. Table 1 shows default maximum buffer size and default TCP socket buffer size at various OSs.

Table 1. Buffer size comparison at various to OSs

Type of Operating System	Default max socket buffer size	Default TCP socket buffer size
FreeBSD 2.1.5	256 Kbytes	16 Kbytes
Linux 2.4.00	64 Kbytes	32 Kbytes
Sun Solaris 7	256 Kbytes	8 Kbytes
MS Win2000 or Win XP	1 Gigabyte	8 Kbytes

Buffer Share Algorithm. At the first stage of TCP Auto-tuning implementation, each connection decides expected socket buffer size by increasing window size appropriately. Memory pool is used to assign resources to each connection. The connection that requires smaller buffer size than "Fair share" reserves expected buffer size. The remaining memory is assigned to connections that require larger size fairly. In order to assign buffer size at next negotiation, this fair share algorithm is configured as "current share."

3 TCP Buffer Management Algorithms

3.1 Buffer_size_negotiator

This research is focus on the Access Grid and the Data Grid, and both they require large data transmission such as gigabytes or terabytes level transmission in Grid application. Grid network affects high speed/high performance network's form, so TCP connection management algorithm is required, which is differentiated with large data transmission method of general internet environment using TCP. In this paper, If large TCP data transmission is generated by Grid application, analysis the characteristic of each TCP connection and compose module that can improve performance by automatic reconfiguration of The TCP buffer size, which appropriate for characteristic of The connection.

Fig. 2. shows buffer_size_negotiator's function on option. Traffic Analyzer achieves function that uses system commands ("netstat" or "lsof") or system information ("/proc" in system) to get information about each TCP connection. These information are entered into "buffer_size_negotiator" by "Information Provider" in Globus and then negotiate the send buffer size of each TCP connections according to setting option. Actually, GridFTP that takes charge Grid application's data transmission applies the negotiated buffer size to each TCP connections. Buffer_size_negotiator has four options as follows;

- default: none
- Max-min fair share: –m
- Weighted fair share: –w
- Minimum threshold: –t

3.2 Buffer_size_negotiator

This section explains buffer management algorithm that corresponds to buffer_size_negotiator's each option. This buffer_size_negociator accepts information form the MDS, which is resource information provider. And then, it negociate each TCP connections' receiving buffer size. The following subsections are the buffer management algorithm provided by buffer_size_negotiator.

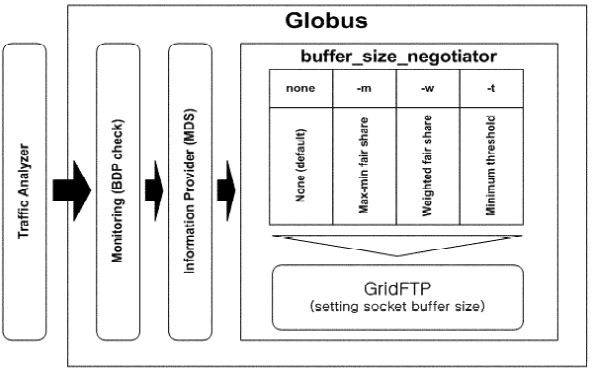

Fig. 2. Buffer_size_negotiator's options

Max-min Fair Share Algorithm. Max-min fair share algorithm is a method that equally establishes re-assignable buffer size to each TCP connection. Table 2 shows the results of the negotiated buffer size using max-min fair share algorithm.

Table 2. Buffer allocation result using –m option

Unit : Kbytes

P	N		Connection A	Connection B	Connection C
1024	3	D(i)	64	1024	768
		B(i)	64	480	480

Weighted Fair Share Algorithm. Weighted fair share algorithm is a method that establishes re-assignable buffer size to be proportional to the requested buffer size of TCP connections. Table 3 shows results of the negotiated buffer size using weighted fair share algorithm.

Table 3. Buffer allocation result using –w option

Unit : Kbytes

P	N		Connection A	Connection B	Connection C
1024	3	D(i)	64	1024	768
		B(i)	64	500	460

Minimum Threshold Algorithm. If one TCP connection's the requested buffer size is smaller than 5% of the requested buffer size's sum of all TCP connections, minimum threshold algorithm guarantees the connection's buffer size to be 5% of all available buffer size (P). Table 4 shows results of the negotiated buffer size using minimum threshold algorithm.

Table 4. Buffer allocation result using –m option

Unit : Kbytes

P	N		Connection A	Connection B	Connection C
1024	3	D(i)	64	1024	768
		B(i)	53	520	451

4 Tests and Analyses

4.1 GridFTP Data Transmission Tests

In this paragraph, the large data transmitting tests using GridFTP are experimented with executed buffer_size_negotiator on each option. We measured the amount of data transmission per second for 3 TCP connections that demand different bandwidth during.

Fig. 3. shows data transmitted amount which measured by GridFTP per second when buffer_size_negotiator configures each option (default, -m, -w and –t).

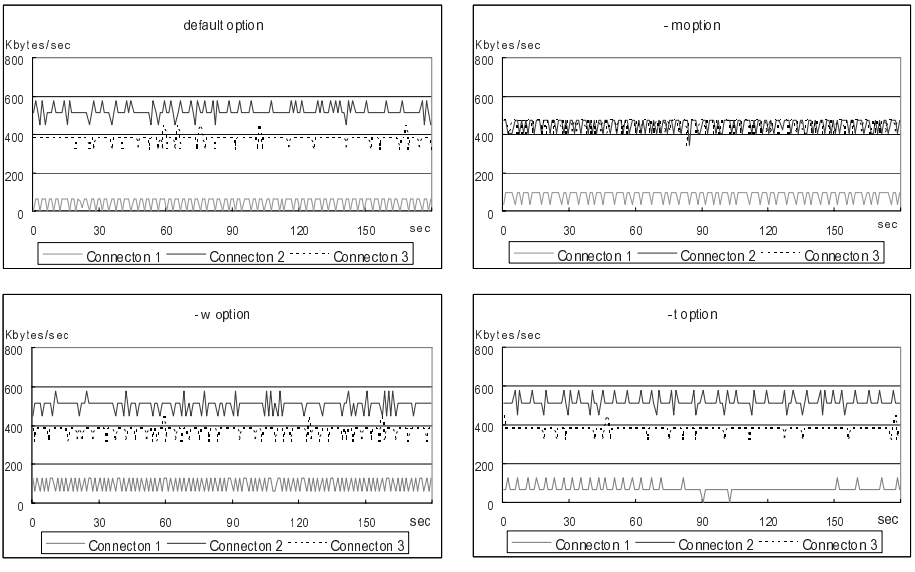

Fig. 3. Results of data transmission test using each option

4.2 Result Analysis

Negotiated Buffer size Comparative Analysis. Through experiment result of 4.1, the requested buffer size of connections and the negotiated buffer size by each buffer management's algorithm are presented at Tab. 5.

Table 5. Comparison of negotiated buffer size using each option

Unit : Kbytes

Option / Conn. name	Requested size	-m option		-w option		-t option	
		Negotiated size	Rate of increase	Negotiated size	Rate of increase	Negotiated size	Rate of increase
Conn. 1	64	64	0 %	64	0 %	53	-17 %
Conn. 2	1024	480	-53 %	500	-51 %	520	-49 %
Conn. 3	768	480	-37 %	460	-40 %	451	-41 %

The following express three kinds of connection features are:

- Conn. 1: TCP connection that requires the small buffer size
- Conn. 2: TCP connection that requires overbalance of maximum buffer size
- Conn. 3: TCP connection that does not exceed maximum buffer size

The buffer size entered into an agreement is established within maximum buffer size when tested TCP connections on each option. When use “-m” and “-w” option, “Conn. 1” allocates the requested buffer size to the negotiated buffer size. However, in case of using ”-t” option, it setablished by 53Kbytes, decreased 17% than the requested buffer size of original.

“Connection 2” and “Connection 3” shows that the negotiated buffer size for all options is established lower than the requested buffer size. There is advantage that “-w” and “-t” options could assign more buffer size for connections that could send large data per second than “-m” option. Also, in case of “-t” option, as sum of the requested buffer size (S) is changed. There is advantage that the negotiated buffer size of the connection which requires small buffer size, could be dynamically changed.

Data Transmission Rate Comparative Analysis. The data transmission amount by GridFTP is same with Table 6 in each option.

Table 6. Comparison of data transmission amount using each option

Unit : Kbytes

Option / Conn. name	None	-m		-w		-t	
		Transmission amount	Rate of increase	Transmission amount	Rate of increase	Transmission amount	Rate of increase
Conn. 1	7,872	14,400	+82.9%	17,664	+124.4%	12,864	+63.4%
Conn. 2	93,760	80,673	-14.0%	90,688	-3.3%	93,440	-0.3%
Conn. 3	67,712	80,736	+19.2%	66,752	-1.4%	68,224	+0.8%
Sum	169,344	175,808	+ 3.8%	175,104	+3.4%	174,528	+3.1%

Such as table 6, if established send buffer size of each TCP connections in GridFTP to the negotiated buffer size by buffer_size_negotiator buffer size, data transfer rate increases in data transmission. Could heighten transfer efficiency of about 3.8% when used "-m" option, and case of 3.4% when used “-w” options and improved performance of 3.1% when used "-t" option

5 Conclusions

In the traditional Internet environment, there was no change in transmission amount of data even on the modified TCP buffer size in case of transmitting large data. On the other hand, the high-performance networks, such as STAR-TAP, KREONET, Grid networks make a profit on the change of TCP buffer size according to network environment. Therefore, Grid Applications in high-performance network is needs dynamic configuration of TCP send/receive buffer size.

This study is improving performance of traffic transfer for GridFTP in grid network of high performance and TCP buffer management for many concurrent TCP connection controls. We implement buffer_size_negotiator for dynamic configuration of send buffer size among TCP connections in GridFTP. In case of transmitting data of The Grid applications by GridFTP, each TCP connection transmits data using buffer size which set by "buffer_size_negotiator." Improved GridFTP performs much better than standard general GridFTP, Which achieves an improvement of 3~4%.

References

1. Hasegawa, T. Terai, T. Okamoto and M. Murata, "Scalable socket buffer tuning for high performance Web servers," Proc. of IEEE ICNP 2001, Nov. 2001.
2. Hethmon, P. and Elz, R., "Feature negotiation mechanism for the File Transfer Protocol", RFC 2389, August 1998.
3. J.Semke, J. Mahdavi, and M. Mathis, "Automatic TCP Buffer Tuning", ACM Sigcomm '98/Computer communications Review, Volume 28, Oct. 1998.
4. Jeffrey Semke, "Implementation Issues of the Autotuning Fair Share Algorithm", PSC Technical Report, May 2000.
5. Qingming Ma, Petter Steenkiste and Huizhang, " Routing High bandwidth Traffic in Max-min Fair Share Networks", ACM Sigcomm '96/Computer communications Review, Volume 26, Aug. 1996.
6. T. Dunigan, M. Mathis and B. Tierney, "A TCP Tuning Daemon", Proceeding of IEEE Supercomputing 2002 Conference, Nov. 2002.
7. V. Jacobson, R. Braden and D. Borman, "TCP Extensions for High Performance", IETF RFC 1323, May.1992.
8. W.Allcock, J.Bester, J.Bresnahan, A.Chervenak, L.Limin, S.Tuecke, "GridFTP: Protocol Extensions to FTP for the Grid", GGF draft, March 2001.
9. `http://www-didc.lbl.gov/TCP-tuning`
10. `http://www.psc.edu/networking/auto.html`, "Automatic TCP Buffer Tuning Research", web page.