Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Improving network service performance and reliability via links trunking technologies*

GUO Hui†[1,2], WANG Yun-peng[2], WANG Zhi-guang[1], ZHOU Jing-li[2]

([1]*Department of Computer Science and Technology, China University of Petroleum, Beijing 102249, China*)

([2]*Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*)

†E-mail: guohui_hust@hotmail.com

Received April 15, 2005; revision accepted Aug. 29, 2005

Abstract: With the increase of high-speed network backbones, the performance of server's network interface gradually becomes a pivotal factor. This study provides a method called Ethernet Links Trunking (ELT) technology for achieving efficient connectivity between backbones and servers, which provides higher bandwidth and availability of server network interface. The overview of the ELT technology and the results of performance experiment are presented in this paper. Findings showed that the network bandwidth can be scaled by multiple ELT technologies so that more reliable network connectivity can be guaranteed. Some crucial techniques such as Adapter Load Balancing (ALB) and Adapter Fault Tolerance (AFT) are presented in this paper. Experimental results showed that parallel channels of Fast Ethernet are both necessary and sufficient for supporting the data rates of multiple concurrent file transfers on file server.

Key words: Ethernet links aggregation, High-speed network, Network performance, Network availability, Adapter load balancing (ALB), Adapter fault tolerance (AFT)

doi:10.1631/jzus.2006.A1001 **Document code:** A **CLC number:** TP302

INTRODUCTION

Because of the critical role that computing systems and networks play in today's business environment, it is not surprising that business success often goes to the company with the best information and the most efficient process for delivering that information. As the principal tool for business collaboration, corporate LANs are a vital resource in enabling today's enterprises to adopt competitive strategies. The number of users wanting to share and access data across enterprise networks and the Internet is increasing dramatically. But increased reliance on corporate networks also means a corresponding increase in network traffic. As the number of clients entering the network at high speeds increase, network administrators now find that their server and network backbones lack the capacity to handle the increased traffic. In addition, network resources are also being strained by the increasing popularity of bandwidth-intensive applications. Today, video and state-of-the-art multimedia applications, with data streams hundreds of megabytes in size, continue to proliferate across corporate LANs, gobbling up an ever-increasing amount of network bandwidth.

To achieve highly efficient connectivity between backbones and servers, and higher bandwidth of server network interface, we provide a method named as Ethernet Links Trunking (ELT) technology. It is a high-speed networking solution that builds upon Fast Ethernet technology to provide a dramatic increase in network performance. Using ELT, network servers have the ability to aggregate multiple Fast Ethernet links into a single logical link to establish a scalable "fat pipe" to carry higher data rates than any single Ethernet link can accommodate. It can be used to

* Project (No. 2001AA111011) supported by the the Hi-Tech Research and Development Program (863) of China

improve the performance and availability of the server network subsystem.

FUNCTIONAL DESCRIPTION

ELT is a port-aggregation technology that aggregates multiple network physical links in parallel to form a single, high-speed logical link so that the throughput between the server and switch can be increased dramatically. It includes two fundamental functions: Adaptor Load Balance (ALB) and Adaptor Fault Tolerance (AFT) technologies.

ALB supports load balancing the distribution of traffic—including unicast, broadcast, and multicast—evenly across the aggregated links and provides higher network I/O performance.

AFT provides redundant parallel paths. If a link is unplugged, damaged, or fails, the AFT software automatically removes the failed link from the port grouping and redistributes the load across the remaining links, without user intervention. As long as one network link connectivity is available, this server is still accessible.

Fig.1 shows the connection between the server and switch when using ELT technology with Fast Ethernet. In the figure, two individual links between a switch and a server are aggregated into one high speed logical link to create a "fat pipe" to relieve congestion in a dedicated area of a network. In addition to providing dramatic increase in network performance, ELT maximizes network availability through its load balancing and redundancy features and enforces all traffic distribution evenly across the aggregated links and AFT provides fault tolerance and failure recovery capability.

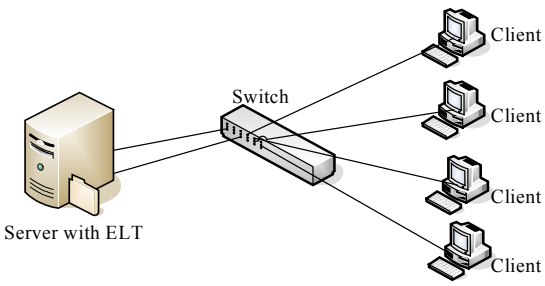

Fig.1 Connection with two links aggregation

DESIGN AND IMPLEMENTATION

System module

These functions are achieved in a user transparent manner through changes in the operating system. We employ FreeBSD (McKusick and Bostic, 1996) as the experimental operating system providing a sophisticated and robust system software platform, together with source code without legal constraints. In our design we chose an Intel pro100+ fast Ethernet card, whose corresponding driver file in FreeBSD OS is the if_fxp.c file in /sys/dev/fxp/ directory. Modifications to the Intel network card driver were made to support ELT. Without modifying any existing protocol and application, we added a middleware driver between the Physical Layer and Media Access Control layer which is transparent to applications from the view of the upper layer. The whole ELT function is realized by four modules. The middleware is comprised of two modules, Adaptor Receive/Send module and Adaptor Switch module. Another two modules—Link Inspector and NIC Scheduler are realized in the User Layer as shown in Fig.2.

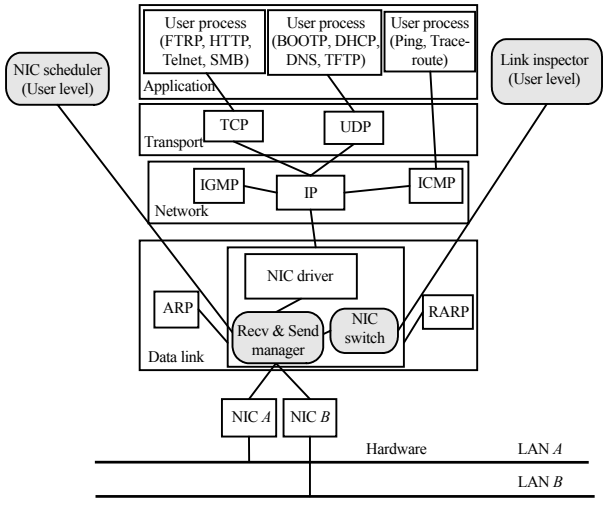

Fig.2 ELT system module

Adaptor Recv/Send Manager is a critical module in ELT implementation and is responsible for intercepting MAC frame passing from the upper layer, and distributing it to respective network adapter with Round-Roll algorithm, rather than a

particular interface. In this way network load is distributed and the system can achieve optimal performance. NIC Switch module realizes network channel transition when the link inspector detects adapter failure. It receives NIC status information passing from Link Inspector and sends command to the Recv/Send Manager to ensure that data packet is always sent from the remaining fine links, for achieving failover function. The Link Inspector module is a background process running at user level, and can inspect link status through Media Independent Interface (MII) information. Together with NIC switch module, ELT achieves failure inspect and recovery character. NIC scheduler provides a set of command interface for user, such as add or delete network card from the aggregated link. User can designate which adapter is to be used for trunking and for working together, or clean all the adapters to make them work separately. Before adding a new card, the scheduler should inspect the connectivity between the card and the switch firstly, and ensure that the adapter has been activated, then configure the new adapter's parameter to be the same as that of the primary adapter, such as IP address, MAC address, netmask, MTU, etc.

Load distributing algorithm

When multiple network adapters are aggregated and working together, how the system determines which one is the current port to send data to? It should be have a choosing mechanism which insure that all the ports have equal network load, this is just the load balancing algorithm (Bryhni *et al*., 2000). Through several years study, many researchers proposed several load balancing algorithms: Round-roll algorithm (Liu *et al*., 2002), MAC address based algorithm (Cisco, 2000), IP address based algorithm (Cao *et al.*, 2000) and TCP port based algorithm (Cardellini *et al.*, 1999) and application-level based algorithm (Hari *et al*., 1999; Sahner *et al*., 1996). In our implementation, we chose Round-Roll algorithm as our load balancing method. Because the link aggregation target was designed for our NAS server system (Guo *et al.*, 2003), redundant calculation on networking is not adapted to NAS hardware system. Round-Roll is a relatively simple algorithm whose implementation is brief, and calculation is relatively more simple. The other algorithms involve too many calculation resources that lead to more delay in the decision-making period of sending frame, so that the whole system performance is degraded. In addition as the algorithm aims directly at the system hardware, network load balancing can always be guaranteed.

In the implementation we established a network adaptors chain, where any adaptors needed for trunking must first be added to this chain, and the IP address of the primary network adaptor is duplicated on all the interfaces in the NIC chain. All the packets received from the upper layer (TCP/IP Layer) are supposed to be sent out from different physical network adaptors alternately by the Round-Roll algorithm. In the Round-Roll algorithm realization, the channel selector specifies a network port for each MAC frame. Beginning with the first network channel, the next frame is sent from the next port, etc. After the last port has been selected, the next round will begin from the originally first one as shown in Fig.3. The Round-Roll algorithm guarantees that the Ethernet packets are evenly distributed over the available network adaptors with the minimum CPU process time, and therefore relieve the network traffic across the respective adaptors. From the client end point of view, the server seems to have only one network adaptor and a single IP address. The modifications of NIC driver method described above are implemented between the Data Link Layer and Physical Layer.

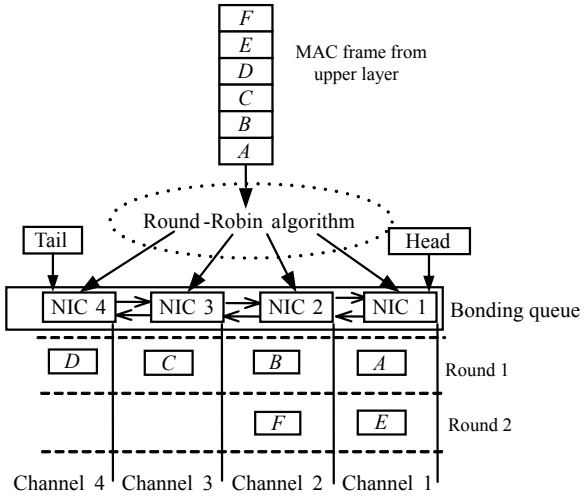

Fig.3 Round-Roll algorithm

The Round-Roll algorithm has load balance property. The AFT implementation used MII tech-

nology by which we can detect the connective between the server NIC and the switch. Before sending out a data package from an interface, we always check the connectivity between this interface and the switch. Only if the connection is fine will the data frame be sent from the specified channel, otherwise the program will choose the next one instead from the NIC chain. In this way the software automatically redistributes the loads across the remaining fine links.

EXPERIMENTAL PERFORMANCE

Network bandwidth

The actual performance of the network interface is a noticeable issue after multiple NICs have been trunking. Based on gigabit test environment, this section presents experimental results before and after link aggregation. The performance of ELT had been tested through Netperf (Hewlett-Packard Company, 1995) benchmark, which can measure the maximal throughput between two hosts. The reason for choosing Netperf instead of other tools such as NetBench is that Netperf is not concerned with disk I/O, but only tests the network interface transmission capacity. So we need not consider the disk I/O bottleneck.

Our testing method establishes interconnection between a server configured with ELT and a gigabit NIC computer. We use LINKSYS EG0801W-8+1 Workgroup Gigabit Switch for connection between the server and the computer. Two network adapters in the server connect to different switch ports respectively, and there is a gigabit-NIC computer connected with this switch on the other end, as shown in Fig.4. We test the network throughput between the two machines. Three conditions were tested: only one NIC was working; two NICs were working simultaneously; and two NICs were present but one of them was off-line (AFT status). The test duration was 10 min.

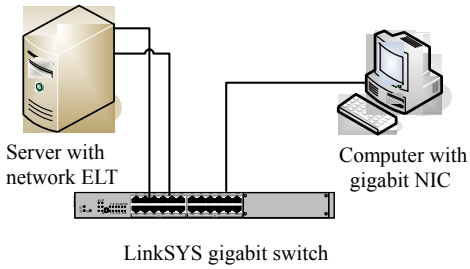

Fig.4 Interconnection between ELT server and gigabit NIC machine

Because the gigabit NIC machine can provide 1000 Mbps bandwidth, the network bottleneck is located in the server with ELT configuration end. Table 1 shows the testing results under three different conditions: one NIC, failover state and trunking state. From this table we can conclude that the server with network link aggregation can provide considerably higher network bandwidth than without the aggregation status. In normal status, the network bandwidth is about 94 Mbps, and the CPU load is 38%. After ELT configuration, the network bandwidth increases to 176 Mbps, which is 1.88 times compared to that before, while the CPU load is still 38%, showing that the Round-Roll algorithm did not increase CPU load, and that the system performance will not be unduly influenced after ELT. Meanwhile, under the AFT condition, the server network performance is approximately the same with one NIC working status. These conclusions are what we expected.

Table 1 Network adapters achievable bandwidth

Test item	Single	AFT state	Two links
Bandwidth (Mbps)	93.91	89.40	176.35
CPU load	38%	39%	38%

File transferring testing

Netperf is designed for examining the maximum achievable throughput capacity of networks. It may be asked whether the dual parallel network is actually valuable under practical conditions such as considering the factor of disk I/O. If so, how about the configuration with three or four adaptors? And under what conditions can the channel trunking's effect be most easily achieved?

To resolve these questions, two experiments were designed to measure the efficiency of the network and disks in the server prototype system. Like the Beowulf test method (Sterling *et al.*, 1995), a synthetic program was used to generate an artificial network traffic to approximate the maximal sustained usable bandwidth of the network. The program consisted of a pair of processes (server process and client process) to exchange a fixed size token (message) between them. One process is executed on server, and

another is executed on client. Server process generates a token comprised of one or more packets and sends it to the client process which receives and stores it in a buffer, and then immediately returns it to the producer. The network load in this experiment is increased by increasing the token size over a range from 4 bytes to 8 kB, and by increasing the number of producer-consumer pairs over a range from 1 to 6. Six producer-consumer pairs means that 6 clients exchange tokens simultaneously with server system running with 6 producer processes. The token exchanges were implemented via standard BSD sockets and the *send*( ) and *receive*( ) system calls under TCP/IP, the system's aggregate network throughput can obtained by FreeBSD netstat command.

This experiment was performed in both one and dual channel configurations. Data bandwidth results are presented in Fig.5 where the number of tokens was varied from 1 to 6 and the token size measured was 2^n bytes where n was varied from 2 to 13 (4 bytes to 8 kB). Experimental results were shown in 12 separate curves divided into two groups. Solid lines represent runs using only one Ethernet and dotted lines show measurements of experiments using dual parallel Ethernet networks. For each of the groups there are 6 curves distinguished by the number of token demand. Fig.5 shows 1~6 active clients connected.

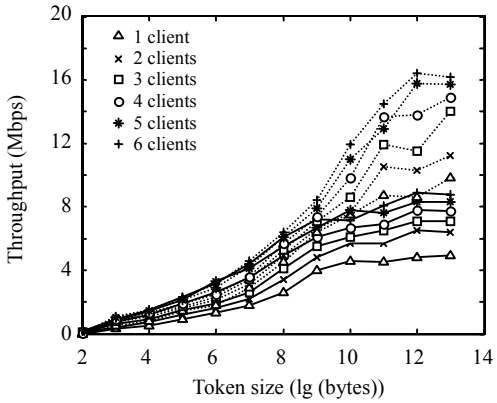

Fig.5 Network throughput test. Solid lines represent runs using only one Ethernet and dotted lines show measurements of experiments using dual parallel Ethernet networks

The data rate has strong relationship to token size but is insensitive to the number of token demands. The maximal achieved throughput in one channel mode was 8.9 Mbps or about 71.2% of the 12.5 Mbps peak for Ethernet. In two channels mode, the maximum throughput was 16.4 Mbps, with 65.6% of the theoretical 25 Mbps peak for dual channels, which is much higher than the peak for a single network channel. From this chart we also can see that when the token size is smaller than 2^9 (512) bytes, the dotted lines are close to solid lines. However, with continued increment of larger token size, the dual channel throughput increased more dramatically than that in single channel mode. It means that only under favorable conditions, larger than 8.5 Mbps network bandwidth, can effective throughput gains be achieved with two multiple concurrent traffic networks. So in the circumstance that the bandwidth of clients request to server is lower than 8.5 Mbps, we cannot benefit from dual Ethernet channel.

Previous experiments only tested the network bandwidth, without considering disk I/O. Are the dual channel trunking configurations necessary and sufficient for data transferring under practical conditions with regard to the aspect of disk I/O bandwidth? To answer this question another test was devised to estimate the disk I/O bandwidth and determine the factors limiting remote file copies. Experimental setup consisted of an ELT server and a computer with gigabit network adapter. The connection method is shown in Fig.4, where all machines have four disks and are organized as RAID 0, which could achieve maximal bandwidth in disk subsystem. In this test, the server reads a fixed size file from all disks, the files are set to be uncached and copy to consumer machine, which can be either the server itself (a local copy) or the gigabit NIC machine (a remote copy). Local file transfers were performed using the POSIX *read*( ) and *write*( ) system calls and a special *sync*( ) call is issued to flush the data blocks on the disks, while the remote file transfers additionally used TCP/IP for transferring the files across the network. We measured the aggregate disk I/O bandwidth of simultaneous file transfers across a mix of local and remote file copies for a range of file sizes. Eight simultaneous file transfers were implemented in this experiment. The disk I/O bandwidth could be obtained by BSD iostat command. To evaluate the influence of network transfers on disk I/O bandwidth, measurements were made with 8 producer consumer pairs for all possible combinations of local and remote file copies with files

size ranging from 1 to 8 MB, and number of remote file copies varying from 0 (all local copies) to 8 (all remote copies).

Fig.6 shows the disk I/O bandwidth as a function of increment of number of remote file copies. The test was run in single channel mode represented by Fig.6a, and in dual channel mode represented by Fig.6b. As we expected, the widest achieved bandwidth of 25.4 MBps occurred when all file copies were local and 1 MB size. The dual and single channel curves shows decreasing trend with increasing remote copy number, but the dual channel trunking curves dipped slightly while the single channel lines dropped dramatically as the remote copy number rose to 4. From this figure, the lowest transfer rate in single channel mode was 6.6 MBps with 8 MB file transfers, while in dual Ethernet channels, the lowest bandwidth was 15.1 MBps. The dual fast Ethernet based system suffers only little degradation when the eight concurrent file transfers vary from entirely local to entirely remote, degrading only about by 18% across the range of tests. So we can conclude that the dual Ethernet throughput imposes few constraints on system bandwidth, and can essentially satisfy the demand of parallel file transfers. The corresponding curves for one channel mode drop dramatically as about 67% of file copies become exclusively remote, due to the constraint by single Ethernet channel. Under favorable conditions the bandwidth gain of two channels with respect to one can reach 2.97. Therefore the dual Ethernet channel is necessary and sufficient for meeting the demands of the parallel disk array throughput and the capacity of communication networks. The statistical results on this experiment are shown in Table 2.

Real applications testing

We use real applications such as FTP transfers, Web browsing and SMB to test the network ability through disarranging packet and removing packet when a link fails. Our finding shows that the FTP and other applications transferring file is normal after ELT is configured. Without resulting problems indicates that the packet ordering can be maintained because of the recombination of TCP/IP layer. Even if

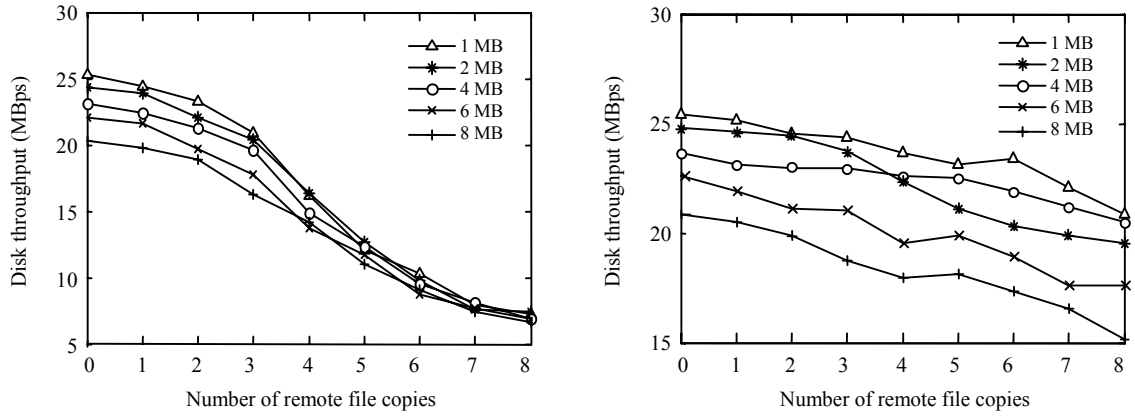

Fig.6 I/O bandwidth with increasing remote file copies. (a) 1 channel mode; (b) Dual channel trunking mode

Table 2 Achievable bandwidth as a function of number of remote file copy

File size (MB)	Single channel mode			Dual channel mode			Improvement (Dual/Single)
	Maximum (Mbps)	Minimum (Mbps)	Drop (%)	Maximum (Mbps)	Minimum (Mbps)	Drop (%)	
1	25.3	7.2	71.4	25.4	20.8	18.1	2.89
2	24.3	7.4	69.6	24.8	19.5	21.4	2.64
4	23.4	6.9	70.5	23.6	20.5	13.1	2.97
6	22.5	6.9	69.3	22.6	17.6	22.1	2.55
8	20.4	6.6	67.6	20.8	15.1	27.4	2.29

the MAC frame sent on link *N* does not arrive before a MAC frame sent on link *N*+1, the upper layer (TCP/IP) protocol can recombine the packets and guarantee them to be ordered. We also tested the network connectivity when one of the ELT channels is cut off. By pulling out a wire manually, we can find that the FTP transferring is in gear, and that the Web page refreshing is fine, which contributes to AFT technique. After inspecting the MII connectivity, ELT automatically redistributes the packets across the remaining fine links, which is so rapid that the upper layer applications cannot perceive it and the file transferring cannot be stopped.

CONCLUSION

ELT technology uses link aggregation to enable network managers to group up two or more Fast Ethernet links (ports) into a single virtual one to obtain higher performance between servers and network backbones. Experimental results proved that ELT is a bandwidth-enhancement technology that can also increase network subsystem availability. Compared with Linux channel trunking technology, ELT implementation is more simple, and additionally provides AFT technique that can bypass a failed link, while the Linux channel trunking technology cannot. Furthermore, the failover switch time is so rapid that the upper-layer applications are not affected. With 4 or more trunking channels, ELT performance is more enhanced. In conclusion, ELT provides low-risk, cost-effective migration path for organizations that require increased performance at the server and backbone levels. Network managers can turn to this links aggregation technology as the solution to network congestion.

References

Bryhni, H., Klovning, E., Kure, O., 2000. A comparison of load balancing techniques for scalable web servers. *IEEE Network*, **14**(4):58-64. [doi:10.1109/65.855480]

Cao, Z.R., Wang, Z., Zegura, Z., 2000. Performance of Hashing-based Schemes for Internet Load Balancing. Proc. the 19th Annual Joint Conference of the IEEE Computer and Communications Societies, p.332-341.

Cardellini, V., Colajanni, M., Yu, P.S., 1999. Redirection Algorithms for Load Sharing in Distributed Web-server Systems. Proc. the 19th IEEE International Conference on Distributed Computing Systems, p.528-535.

Cisco (Cisco EtherChannel Technology), 2000. Cisco System White Paper. Http://www.cisco.com/warp/public/cc/techno/lnty/etty/fsetch/tech/fetec_wp.pdf.

Guo, H., Zhou, J.L., Yu, S.S., 2003. HUSTserver: implementation for reliable and high-performance network attached storage system. *Journal of Shanghai University (English Version)*, **7**(2):156-162.

Hari, A., Varghese, G., Parulkar, G.M., 1999. An architecture for packet-striping protocols. *ACM Transactions on Computer Systems*, **17**(4):249-287. [doi:10.1145/329466.329471]

Hewlett-Packard Company, 1995. Netperf: A Network Performance Benchmark Revision 2.0. Hewlett-Packard Company, p.3-32. Available at http://www.netperf.org.

Liu, J., Kit, H.C., Hamdi, M., 2002. Stable Round-roll Scheduling Algorithms for High-performance Input Queued Switches. Proc. the 10th Symposium on High Performance Interconnects, p.43-51.

McKusick, M.K., Bostic, K., 1996. The Design and Implementation of the 4.4 BSD Operating System. Addison Wesley, http://www.freebsd.org.

Sahner, R.A., Trivedi, K.S., Puliafito, A., 1996. Performance and Reliability Analysis of Computer Systems: An Example Based Approach Using the SHARPE Software Package. Kluwer Academic Publishers, Boston.

Sterling, T., Savarese, D., Becker, D., Fryxell, B., Olson, K., 1995. Communication Overhead for Space Science Applications on the Beowulf Parallel Workstation. Proc. the 4th IEEE Symposium on High Performance Distributed Computing (HPDC), p.23-30.