Evaluation of Gigabit Ethernet with Java/HORB

Y.Yasu†,

H.Fujii, Y.Igarashi, E.Inoue, H.Kodama, A.Manabe, Y.Watase††,

Y.Nagasaka†††, M.Nomachi††††,

S.Hirano, H.Takagi†††††, K.Shudo††††††,

T.Arai†††††††, L.Sarmenta††††††††,

C.Thomborson, R.Nicolescu and M.Duc†††††††††

†HORB Open and High Energy Accelerator Research Organization (KEK), 1-1 Oho, Tsukuba, Ibaraki 305, Japan
††High Energy Accelerator Research Organization, 1-1 Oho, Tsukuba, Ibaraki 305, Japan
†††Department of Electrical Engineering, Nagasaki Institute of Applied Science (NIAS), 536 Aba-machi, Nagasaki, 851-01 JAPAN
††††Research Center for Nuclear Physics(RCNP), 10-1 Mihogaoka, Ibaraki OSAKA 567-0047 JAPAN
†††††HORB Open and Electrotechnical Laboratory (ETL), 1-1-4 Umezono, Tsukuba 305, Japan
††††††HORB Open and MuraokaLab., Department of Information and Computer Science, Waseda U., 1-104 Totsuka-machi, Shinjuku, Tokyo, 169-8050
†††††††NetOne Systems Co.,Ltd., 2-2-8 Sphere Tower Tennoz, Higashi Shinagawa, Shinagawa-Ku, Tokyo 140-8621 Japan
††††††††HORB Open and Laboratory for Computer Science, MIT, 77 Massachusetts Avenue, Cambridge, MA 02139, U.S.A.
†††††††††HORB Open and Computer Science Department, Auckland U., Private Bag 92109, Auckland, New Zealand

Abstract. We have evaluated a high speed network, Gigabit Ethernet(GbE), with Java/HORB, which means Java and Java-based Distributed Object Technology(DOT). Next generation of data acquisition(DAQ) needs high speed network such as ATM and GbE for data transfer in Level 3 and/or Level 2 trigger of the DAQ at large scale DAQ system like Large Hadron Collider(LHC). When evaluating the basic parameters of GbE, we considered bottleneck of network performance such as TCP buffer size, memory access speed, Maximum Transmission Unit(MTU) and so on. We used network tools called TTCP and Netperf and some Java benchmark programs for evaluating DOTs, namely, HORB, RMI and Voyager. Linux and Windows/NT operating systems

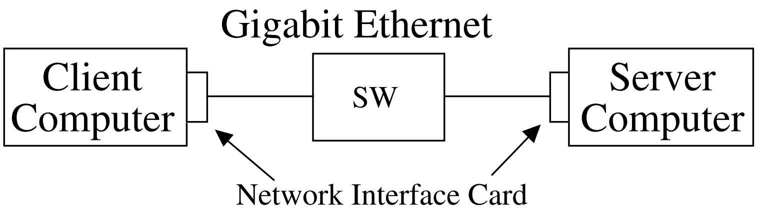

FIGURE 1. Setup for performance evaluation

on PC computers, and Solaris on UltraSPARC workstation were used. MTU had an important role of the data transfer. When 2 Ultra30/Solaris systems via GbE were used, the speed with MTU of 9000 bytes was over 500 Mbit/s(over 60 MB/s) and twice faster than that of 1500 bytes. In evaluation of remote method call as DAQ message path, HORB performance was twice faster than that of RMI and 3 times faster than that of Voyager. In the byte array transfer as DAQ data path, HORB performance was twice faster than that of RMI and Voyager. HORB Serialization was also twice faster than built-in JDK Serialization.

INTRODUCTION

This paper describes performance evaluation of TCP/IP with GbE [1] and Java/HORB [2–5] with GbE.

Platform Independence. To extend life time of software for DAQ, the software needs to be independent of CPU and Operating System(OS) because the large DAQ system requires heterogeneous computer systems. So far, UNIX and TCP/IP were sometimes chosen as standard and Application Interfaces(APIs) of the DAQ were defined. Now there is Java. Java is a key technology of the next generation of DAQ. Java is a pure object oriented language and is architecture-neutral. A Java compiler generates a bytecode which is independent of CPU and OS. The standard Application Interface (API) is available on many computers and the application program will be truly independent of CPU and OS. This is a great feature of Java.

Necessity of Middle-ware The next generation of DAQ also needs network-based DAQ. The DAQ has to manage many computers connected to a complicated network. This increases the software complexity, which causes high cost and low quality. The DAQ system should be simple, reliable and robust. Reliable network programming leads us to middle-ware based on TCP/IP. Distributed Object Technology (DOT) is a form of middle-ware based on TCP/IP and object oriented languages. Java benefits from the existence of DOTs. HORB is the most prominent DOT.

WEB Computing. A web browser such as Netscape Navigator will play a major role in the DAQ client functions such as run-control. A Java applet, which is a program tied to the web browser, can run on any browser. This feature of Java is also important.

Java Solution. We have already proposed that we would establish a 3-tier model of DAQ [6]. The model defines a DAQ client, a DAQ server and a DAQ database.

It emphasizes that all of the programs should be based on Java.

Evaluation of Gigabit Ethernet(GbE). Next generation of DAQ needs high speed network such as ATM and GbE for data transfer in Level 3 and/or Level 2 trigger of the DAQ at large scale DAQ system. Why do we evaluate GbE? Ethernet is an international standard in local area network. GbE is being designed to offer high performance and maintains compatibility with current Ethernet. Therefore, GbE promises to meet current networking needs effectively and economically. Then, the cost performance of GbE can be expected to be cheaper than that of ATM. Quality of Service(QoS) is also important for the DAQ because of the guaranteed data flow. Currently GbE does not support the QoS, but various QoSs are discussed and it can be expected in future.

TABLE 1. Computer systems

Computer	Gateway PC	Gateway PC 1	Fujitsu PC	Sun Ultra30
CPU	PentiumII 266MHz	PentiumII 300MHz	PentiumII 333MHz	UltraSPARC-II 300MHz
Memory	64MB	64MB	190MB	128MB
Bus	PCIbus 133MB/s	PCIbus 133MB/s	PCIbus 133MB/s	System bus 1.6GB/s with PCIbus

System	WNT	WNT1	Linux	Linux1	Solaris
Computer	Gateway	Fujitsu	Gateway	Gateway 1	Ultra30
OS	WindowsNT4 SP3	WindowsNT4 SP3	Linux2.0.29	Linux2.0.30	Solaris2.6
Java	JDK1.1.6 SunJIT,MSJIT	JDK1.1.6 SunJIT,MSJIT	JDK1.1.5	JDK1.1.5	JDK1.1.6 SunJIT
C compiler			gcc 2.7.2.1	gcc 2.7.2.1	gcc 2.7.2.3

TABLE 2. Network equipments

NIC	Packet Engine (GNIC)	Alteon (AceNIC)

SW Bandwidth of Backplane	Alteon AceSwitch180 8 Gps	NetOne ProminetP550 41.6 Gps

I TCP/IP PERFORMANCE EVALUATION

In Table 1, computer systems we used for the performance evaluation were shown. The setup for the evaluation is shown in Fig. 1. There are two computer systems at the measurement, namely, a client system and a server system. We tested

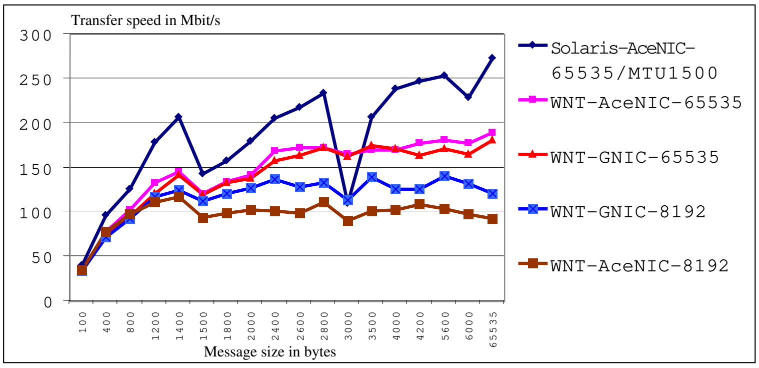

FIGURE 2. TCP/IP Performance with Netperf : MTU = 1500

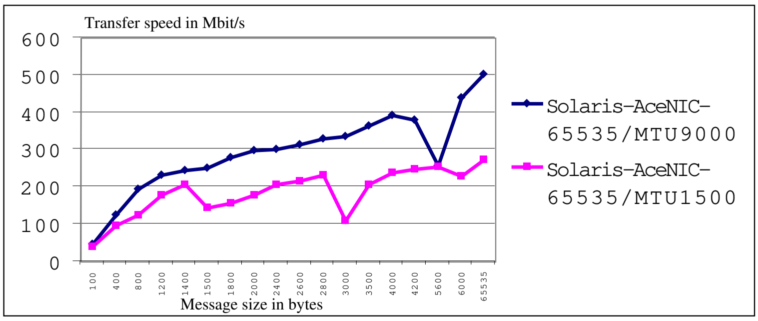

FIGURE 3. TCP/IP Performance with Netperf : MTU = 9000

three combinations of computer systems. One is Windows/NT systems, WNT as client and WNT1 as server. Another is Linux systems, Linux as client and Linux1 as server. The other is Solaris systems, 2 Solaris as client and server. In Table 2, Network Interface Card(NIC)s and Switches are shown. Alteon SW supports large Maximum Transmission Unit(MTU). The performance evaluation using large MTU could be done only on Solaris systems . When we evaluate the GbE, we have to consider bottleneck of the system like TCP/IP protocol, System bus such as PCIbus and memory access performance, CPU performance and NIC & network switch. We emphasize that MTU is very important parameter to improve the network performance. We checked whether Java and HORB become bottleneck or not.

First, we checked TCP buffer size. We used Windows/NT systems with GNIC and NetOne SW. We used a network tool called Netperf [7]. When the buffer size was 8192 and message size was 8000, the average transfer speed was 17 MB/s. But, the speed became 22 MB/s when the buffer size was 65535 and the message size was 65535. This means the TCP buffer size improved about 30 % of the network performance.

Fig.2 shows results from several combinations. WNT-GNIC-8192 means the systems used WNT/WNT1 and GNICs, and the TCP buffer size was 8192. WNT-GNIC-8192 was 20 to 30 % faster than WNT-AceNIC-8192. This shows the network performance depends on NIC. Fig.2 also shows that the network performance

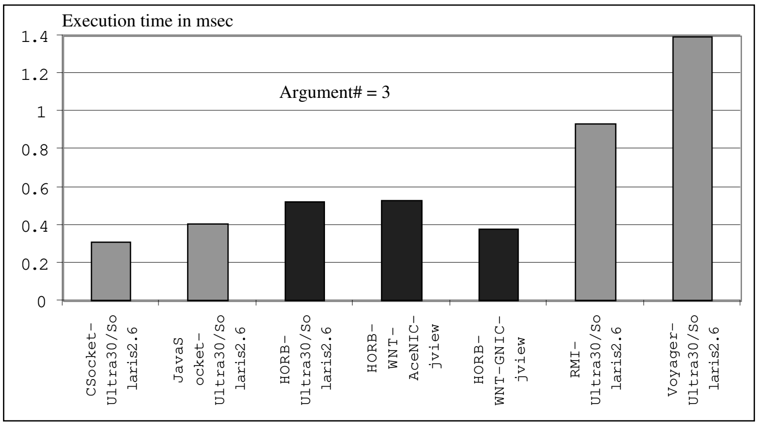

FIGURE 4. DOT Performance : remote method call

depends on system bus and memory access performance. Because Solaris-AceNIC-65535 was up to 50 % faster than WNT-AceNIC-65535. The Solaris system has faster system bus such as 1.6GB/s. We measured the memory access speeds of Solaris systems and WNT/WNT1 systems, by using our own benchmark program. As a result, the speed of the Solaris was twice faster than that of the WNT. The CPU performance of the Solaris is similar to that of the WNT in comparison with SPECint95.

As described before, Alteon SW supports large MTU. We evaluated the performance with MTU=1500 and MTU=9000 on the Solaris systems. When MTU enlarges, data size per transfer on Ethernet frame enlarges. Fig.3 shows that large MTU improved the network performance twice. Previous parameters like TCP buffer size improved the network performance, but up to a few ten %. MTU improved 100 % of the network performance. This means large MTU is necessary for improving the performance dramatically.

Finally, we checked CPU usage at data transfer. We used a tool called TTCP [8]. We assume that TCP buffer size was 65535 and message size was 10000. When MTU was 1500, the transfer speed was also 22 MB/s on Linux/Linux1 systems, not only on WNT/WNT1 systems. At the measurement, the client program consumed 33 % of CPU time for the data transfer while the server program consumed 24 % of the CPU time. On Solaris systems, when MTU was 1500, the speed was 34 MB/s. The CPU usage on the client was 50 % and that on the server was 62 %. On the other hand, when MTU was 9000, the speed was 56 MB/s and the CPU usage on the server became 83 %. Obviously, the CPU usage on the server increased when the transfer speed increased. The CPU performance may be bottleneck when the speed increases. It is due to TCP/IP protocol. New implementation such as zero-copy driver will be necessary.

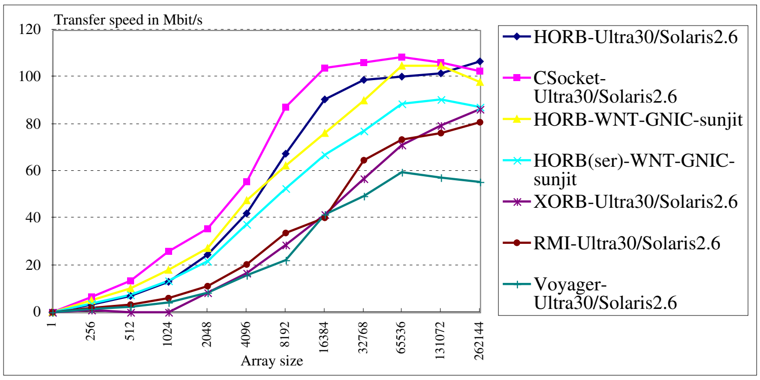

FIGURE 5. DOT Performance : byte array transfer with MTU = 1500

II DOT PERFORMANCE EVALUATION

The Distributed Object Technologies(DOTs) we evaluated are shown below.

- HORB 1.3.b3
- HORB Serialization Patch 1.1a [9]
- RMI (JDK1.1.6) [10]
- Voyager2.0 beta2 [11]
- XORB [12] (an experimental ORB running on JDK1.2 beta3)
- DOT emulator with C socket
- DOT emulator with Java socket

We already evaluated the popular DOTs [2,3]. After that, not only HORB but also RMI and Voyager were improved. We evaluated remote method call of DOT for message path of DAQ and the byte array transfer for data path of DAQ. We also evaluated performance of object serialization of HORB and JDK.

Remote Method Call. Fig.4 shows result from the remote method call. C socket was the fastest. Next was Java Socket. HORB was twice faster than RMI and three times faster than Voyager. As described above, network performance depends on NIC. When NIC was changed, HORB performance was improved , and relative speed of HORB to C socket and Java socket was also improved. We can say HORB performance as DAQ message path is similar to C performance. Thus, HORB will be used as message path of DAQ.

Byte Array Transfer(MTU=1500). There are two results from byte array transfer. Fig.5 shows DOT performance in MTU of 1500 bytes and Fig.6 shows that of 9000 bytes. C socket was also the fastest, but C socket, Java socket and HORB had similar performance. HORB was twice faster than Voyager and several ten % faster than RMI. It seems that the performance was saturated up to 110 Mbit/s.

Byte Array Transfer(MTU=9000). When MTU was 9000, the situation changed. HORB performance was twice improved at large array size. Voyager and RMI were also improved.

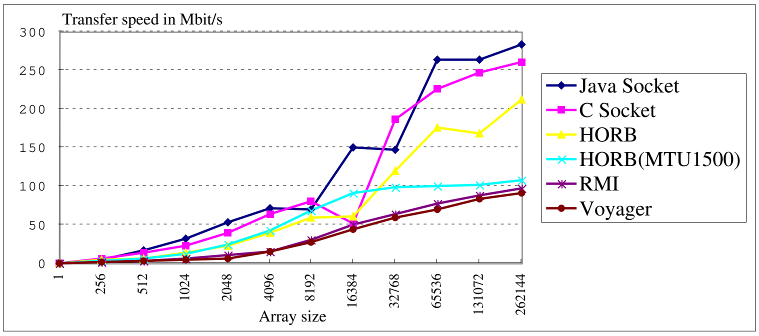

FIGURE 6. DOT Performance : byte array transfer with MTU = 9000

Object Serialization. A reason why HORB is faster than RMI, Voyager, and other DOTs is that HORB's serialization mechanism is faster than JDK's built-in serialization. Fig. 5 shows that the performance of XORB was same as that of RMI. XORB has simple protocol for the transfer while RMI does not have the special protocol such as connection-multiplex. HORB Serialization Patch was slower than the original HORB for the transfers. Fig. 7 shows results for transferring the objects shown in Listing 1. DataArr is an object containing an array of Data objects, each of which wraps an integer. DataChain is a linked list of integers.

```
public class DataArr
 implements java.io.Serializable {
 Data[] da;

 public DataArr(){ }
 public DataArr( int size ) {
  da = new Data[size];
  for ( int i = 0; i < size; i++ )
    da[i] = new Data(); }
 }
```

```
public class DataChain
 implements java.io.Serializable {
 int a;
 DataChain next;
 public DataChain() { }
 public DataChain( int size ) {
  // create chain of size elts
  if ( size > 1 )
   next = new DataChain(size-1); }
}
```

Listing 1: DataArr and DataChain classes for object transfer benchmarks

Here we see that the original HORB performed much better than RMI and HORB with JDK serialization, indicating that JDK Serialization is slower than HORB's serialization. From these results, we can conclude that JDK Serialization is a bottleneck in other DOTs, and predict that the original HORB will perform better than *any* DOT that uses JDK Serialization.

III CONCLUSION

TCP Performance with GbE. At large TCP buffer the network performance was 30 % improved. The difference of network performance at small TCP buffer existed by different NIC. When memory access speed became twice, the network performance was up to 50 % improved. When MTU was 9000, the speed was twice faster

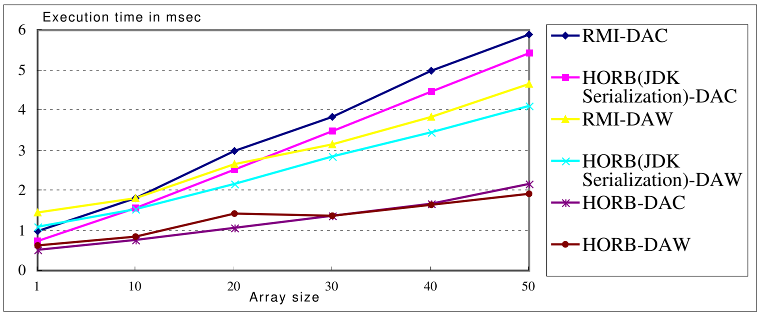

FIGURE 7. DOT Performance : object transfer

than that when MTU was 1500. When network performance increases more and more, new implementation such as zero-copy driver will be necessary for reducing CPU usage.

DOT Performance with GbE. Message path of DAQ was evaluated by the remote method call. HORB can have similar performance to C. Data path of DAQ was also evaluated by the byte array transfer. HORB had good performance at large array size on byte array transfer, but HORB performance like other DOTs was very poor at 'int' and 'double' array transfers. HORB is now the fastest in popular DOTs. HORB's serialization is faster than JDK serialization, so we can predict that HORB will perform better than *any* ORB based on JDK Serialization.

REFERENCES

1. Gigabit Ethernet Alliance. http://www.gigabit-ethernet.org/
2. HIRANO, S., HORB: Distributed Execution of Java Programs, Worldwide Computing and Its Applications, Springer Lecture Notes in Computer Science 1274 pp.29-42, 1997
3. S.Hirano, Y.Yasu and H.Igarashi, Performance Evaluation of Popular Distributed Object Technologies for Java, ACM1998 workshop on Java for High-Performance Network Computing, Palo Alto, California, February 28 & March 1, 1998.
4. HORB Homepage. http://ring.etl.go.jp/openlab/horb/
5. HORB Open Homepage. http://www.horb.org/
6. Y.Yasu el al.,IEEE Trans. on Nucl. Sci., Vol45,No.4,August 1998
7. Netperf Homepage. http://www.netperf.org/netperf/NetperfPage.html
8. ftp site of TTCP. http://ftp.arl.mil/ftp/pub/ttcp/
9. Homepage of HORB Serialization Patch.
   http://www.cag.lcs.mit.edu/bayanihan/horb/Serialization.html
10. Sun Microsystems, Remote Method Invocation Specification, 1997
11. Voyager Homepage. http://www.objectspace.com/products/voyager/index.html
12. XORB Homepage. http://java-house.etl.go.jp/~ takagi/java/bit-xorb/
13. JavaDAQ Homepage. http://www-online.kek.jp/~ online/JavaDAQ/