Tru64 UNIX Best Practice

Tuning for Optimal Gigabit Ethernet Performance

May 2001

Product Version: **Tru64 UNIX**

This Best Practice describes guidelines for achieving optimal Gigabit Ethernet performance on Tru64 UNIX systems with the DEGPA PCI/1000 Mbps Ethernet network interface card (NIC).

Compaq Computer Corporation
Houston, Texas

Contents

Tuning for Optimal Gigabit Ethernet Performance

Tuning for Optimal Gigabit Ethernet Performance

Recent testing by the Tru64 UNIX Base Networking group using the Netperf tool shows that for the best Gigabit Ethernet performance, larger is better: larger transfer sizes, larger socket buffer sizes, and larger Maximum Transmission Unit (MTU) sizes. This best practice discusses the results of this testing and gives you guidelines that you as system administrator can use in order to achieve the best Gigabit Ethernet performance for your system.

See the Tru64 UNIX Best Practices Web page for more information about Best Practices documentation.

Is This Best Practice Right for You?

Not all Best Practices apply to all configurations, so you must be sure that it is appropriate for your system and circumstances. To use this Best Practice, you must meet the requirements described in the following table:

Requirement	Description
Operating System	All versions
System Configuration	The system must contain at least one DEGPA PCI/1000 Mbps network interface card (NIC) and connected to a network.
Impact on Availability	None.
Skills	You must have network and system administration skills and experience with system tuning.

Note

Although this Best Practice applies to all versions of the operating system, any links to Tru64 UNIX documentation are to Version 5.1.

If you do not meet the previous requirements, see *Alternative Practices* for information.

Before You Begin

Before you apply the Best Practice for tuning your system for optimal Gigabit Ethernet performance, you must understand the following background information.

- The factors that affect Gigabit Ethernet performance
- The test description and methodology
- The results of recent testing by the Tru64 UNIX Base Networking group
- The various methods of modifying system attributes
- The `ifconfig` command

Factors Affecting Performance

The following factors affect Gigabit Ethernet performance:

Transfer size

The size of the data packet or data block size that an application uses. The application determines this value. You can only increase the transfer size by modifying the application code.

Socket buffer size

The size of the send and receive buffers that a socket uses for a connection. For a Tru64 UNIX Version 5.0 and later systems the default is 61440 bytes. For earlier versions of the operating system, the default is 32768 bytes.

MTU size

The maximum size of a datagram in a network. For Ethernet networks the default is 1500 bytes. For systems with a DEGPA NIC that are attached to a Gigabit Ethernet network, the MTU size is 9000 bytes. This is also called a **jumbo frame**.

Test Description and Methodology

Netperf is a public domain test package available at **`http://www.netperf.org/netperf/NetperfPage.html`**. It includes a robust set of tests and test parameters that enable you to

characterize the performance of a network subsystem with TCP/IP. The Tru64 UNIX Base Networking group used the TCP STREAM test, which is a socket-based, memory-to-memory data throughput test (transmit and receive). The testing was performed with two AlphaServer ES40 systems (EV6 500 MHz CPUs) running Tru64 UNIX Version 5.1 (revision 732) with the PK2 patch kit installed. These systems were connected to an isolated Alteon ACEswitch 180 switch.

In order to obtain useful TCP/IP network performance data, we did the following:

- Used a reasonable, stable, and controlled test configuration. In this case, it means using a pair of identical systems on a private interconnect.
- Used test software designed for performance characterization. In this case, Netperf.

FTP is not suitable for performance measurement or characterization because it uses smaller transfer sizes and non-optimized transmit and receive algorithms. In addition, even if you use a memory file system (MFS), the file system code path is involved.

Testing Results on Tru64 UNIX Version 5.1 with Patch Kit 2

Table 1 shows what happens to throughput when you increase the transfer size while holding the socket buffer size and MTU size constant. The socket buffer size and MTU sizes are the default system values.

Effects of Transfer Size Alone on Throughput

Transfer Size (KB)	Socket Buffer Size (bytes)	MTU Size (bytes)	Throughput (Mbps)
.128 (128 bytes)	61440	1500	167
1	61440	1500	304
8	61440	1500	401
32	61440	1500	423
64	61440	1500	425
256	61440	1500	425

Although increasing the transfer size results in increased throughput, the throughput gains stop and level off.

Table 2 shows what happens to throughput when you increase the socket buffer size while holding a large transfer size and default MTU size constant.

Effects of Socket Buffer Size and Large MTU Size on Throughput

Transfer Size (KB)	Socket Buffer Size (bytes)	MTU Size (bytes)	Throughput (Mbps)
64	32768	1500	339
64	61440	1500	425
64	131072	1500	540
64	262144	1500	545
64	524288	1500	550

Although using a large transfer size and increasing the socket buffer size results in increased throughput, the throughput gains lessen and begin to level off.

Table 3 shows what happens to throughput when you increase the socket buffer size while holding a large transfer size and large MTU size (jumbo frames) constant.

Effects of Large Transfer Size, Socket Buffer Size, and MTU Size on Throughput

Transfer Size (KB)	Socket Buffer Size (bytes)	MTU Size (bytes)	Throughput (Mbps)
1	32768	9000	195
32	32768	9000	253
64	32768	9000	292
64	61440	9000	407
64	131072	9000	771
64	262144	9000	960
64	524288	9000	990

The combination of large transfer size, large socket buffer size, and large MTU size (jumbo frames) is necessary to drive the line to saturation. Using jumbo frames by themselves will not accomplish this.

Modifying System Attributes

You can use the following methods to modify system attributes that affect socket buffer size for all applications on a system:

- `dxkerneltuner` — This is the Kernel Tuner GUI. Access the GUI through the Common Desktop Environment (CDE) Application Manager window; select the System_Admin icon and then select the MonitoringTuning icon. Choose the subsystem whose attribute you want to modify, and enter the new value in the Current Value field. See `dxkerneltuner`(8) for more information.
- `sysconfig -r` — This is a command line interface. Use the following command syntax to modify a system attribute:

  ```
  # sysconfig -r subsystem attribute=value
  ```

  See `sysconfig`(8) for more information.

Alternatively, you can use the `setsockopt` system call to change the options that affect socket buffer size for individual applications. See `setsockopt`(2) for more information.

The ifconfig Command

The `ifconfig` command configures and displays network interface parameters. Use the following command syntax to change the MTU size for an interface:

```
# ifconfig interface-id ipmtu value
```

See `ifconfig`(8) for more information.

Applying the Best Practice

Before you tune your system for optimal Gigabit Ethernet performance, be sure to follow the recommendations in *Before You Begin*.

The following procedure suggests a method of tuning your system and applications to achieve the best Gigabit Ethernet performance:

1. Increase the transfer size to 64 kilobytes (KB) by modifying your application source code, if possible.

2. Increase the socket buffer size to a larger value (for example, 524288) by doing either of the following:

   a. Modify your application source code, if possible, to change the socket buffer size for this application only. Use the `setsockopt` system call and change the SO_SNDBUF and SO_RCVBUF options to use a larger value (for example, 524288).

   b. To increase the default system socket buffer sizes, set value for the `tcp_sendspace` and `tcp_recvspace` system attributes in the `inet` subsystem to 524288 bytes. The following example shows how to do this using the `sysconfig` command:

   ```
   # sysconfig -r inet tcp_sendspace=524288
   # sysconfig -r inet tcp_recvspace=524288
   ```

3. Increase the MTU size to use jumbo frames by issuing the following command. The interface ID in the example is `alt0`.

   ```
   # ifconfig alt0 ipmtu 9000
   ```

   The preceding command is only in effect until the next time the system is booted. If you want have this MTU size to take effect each time the system is booted, edit the `/etc/inet.local` file and put the appropriate `ifconfig` command line into it.

Note

Although you must perform each of the preceding steps in order to get optimal Gigabit Ethernet performance, you can improve performance by performing just one or two of the steps.

Verifying Success

After you apply the Best Practice for tuning for optimal Gigabit Ethernet performance, you can verify whether it was successful by doing the following:

1. Verify that the MTU size has changed by using the `netstat -i` command. See `netstat(1)` for more information.
2. If you modified the global socket buffer sizes, verify that the `tcp_sendspace` and `tcp_recvspace` system attributes in the `inet` subsystem have changed by issuing the following command:

   ```
   # sysconfig -q inet
   ```

If the Best Practice was not successful, see *Troubleshooting* for information about identifying and solving problems.

Troubleshooting

Not applicable.

Alternative Practices

Not applicable.

Comments and Questions

We value your comments and questions on the information in this document. Please mail your comments to us at this address:

best_practices@zk3.dec.com

Legal Notice