

Tru64 UNIX Best Practice

Primary Tuning Recommendations for Internet Servers

February 2002

Product Version: **Tru64 UNIX Version 5.1A or earlier**

This Best Practice describes how to tune Compaq Tru64 UNIX, Version 5.1A and earlier to improve the performance of Internet servers, including Web servers, ftp servers, mail servers and relays, proxy servers, caching servers, gateway systems, and firewall systems.

Contents

Primary Tuning Recommendations for Internet Servers

Is This Best Practice Right for You?	1
Before You Begin	1
Applying the Best Practice	2
Modifying Internet Attributes	3
Increasing the Size of the TCP Hash Table	3
Disabling PMTU Discovery	3
Increasing the Number of Outgoing Connection Ports	4
Modifying Process Attributes	4
Increasing the Size of System Tables and Data Structures	5
Increasing the Number of Processes per User	6
Increasing the Number of Threads per User	6
Increasing the User Process Data Segment Size Limits	6
Increasing the User Process Address Space Limits	6
Modifying Socket Attributes	7
Increasing the Maximum Number of Pending TCP Connections	7
Increasing the Minimum Number of Pending TCP Connections	7
Enabling the mbuf Cluster Compression	8
Modifying Virtual Memory Attributes	8
Increasing the Maximum Number of Memory-Mapped Files	9
Increasing the Maximum Amount of Valid Virtual Address Space	9
Troubleshooting	9
Comments and Questions	10
Legal Notice	10

Primary Tuning Recommendations for Internet Servers

These recommendations result from testing Tru64 UNIX systems running Internet server software such as AltaVista (www.altavista.com). Not all recommendations are appropriate for all types of systems.

See the *Tuning Tru64 UNIX for Internet Servers* document for more information on tuning Internet servers.

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

http://www.tru64unix.compaq.com/docs/best_practices/index.html

Is This Best Practice Right for You?

Not all Best Practices apply to all configurations, so you must be sure that this Best Practice 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	Tru64 UNIX Version 5.1A or earlier
System Configuration	Internet servers, including Web servers, ftp servers, mail servers and relays, proxy servers, caching servers, gateway systems, and firewall systems
Impact on Availability	Changes to some attributes require a system reboot

Before You Begin

Before you apply this Best Practice, you must understand some background information and perform some preliminary tasks.

Attribute Name Equivalence and Tuning Requirements lists the attribute name equivalence for Tru64 UNIX Version 5.0 or higher, and Tru64 UNIX

Version 4.0G and earlier. Although Tru64 UNIX Version 5.0 and higher offers backward compatibility, Compaq recommends that you use the new names for that version of the operating system. This table also lists whether the attributes can be tuned at run time. If Yes is specified in the table, you can change the attribute value at run time; if No is specified, you need to reboot your system to change the attribute value.

Attribute Name Equivalence and Tuning Requirements

Tru64 UNIX Version 4.0G and earlier	Tru64 UNIX Version 5.0 and higher	Tune at Run Time?
ipport_userreserved	ipport_userreserved	Yes
max-per-proc-address-space	max_per_proc_address_space	No
max-per-proc-data-size	max_per_proc_data_size	No
max-proc-per-user	max_proc_per_user	No
max-threads-per-user	max_threads_per_user	No
maxusers	maxusers	Yes for Tru64 UNIX Version 5.0 and higher. No for Tru64 UNIX Version 4.0G and earlier.
pmtu_enabled	pmtu_enabled	Yes
sbcompress_threshold	sbcompress_threshold	Yes
somaxconn	somaxconn	Yes
sominconn	sominconn	Yes
tcbhashsize	tcbhashsize	Yes

Some older versions of Tru64 UNIX do not support some attributes. Others restrict the methods that you can use to modify and display attributes, or they require operating system patches to use the attributes. See the *System Tuning and Configuration* manual for your version of the operating system for more information on these issues.

Applying the Best Practice

Before you apply this best practice, be sure to follow the recommendations in *Before You Begin*.

The operating system includes various subsystems that are used to define or extend the kernel. Kernel subsystem attributes are used to set kernel variables, which control subsystem behavior or track subsystem statistics. Attributes are assigned default values at boot time. For Internet servers,

the default values of some attributes may not be appropriate, so you must modify these values to provide optimal performance.

There are many kernel subsystem attributes that affect Internet server performance. Internet servers include Web servers, ftp servers, mail servers and relays, proxy servers, caching servers, gateway systems, and firewall systems. The following sections describe the tuning recommendations that provide the best performance improvement for most Internet server configurations.

Modifying Internet Attributes

You may be able to improve Internet server performance by tuning the following Internet (`inet`) subsystem attributes:

- `tcblhashsize` (*Increasing the Size of the TCP Hash Table*)
- `pmtu_enabled` (*Disabling PMTU Discovery*)
- `ipport_userreserved` (*Increasing the Number of Outgoing Connection Ports*)

Increasing the Size of the TCP Hash Table

The `tcblhashsize` attribute specifies the number of buckets in the Transmission Control Protocol (TCP) `inpcb` hash table. The kernel must look up the connection block for every TCP packet it receives; therefore, increasing the size of the table can speed up the search and improve performance.

However, increasing the size of the hash table will cause a slight increase in wired memory. It can also cause a bottleneck at the TCP hash table in SMP systems.

The default value is 512 buckets (32 on systems running a version of Tru64 UNIX prior to Version 4.0E). The *recommended value* is 16384.

Disabling PMTU Discovery

Packets transmitted between servers are divided into equal-sized units to facilitate the transmission of the data over routers and small-packet networks, such as Ethernet networks.

When the `pmtu_enabled` attribute is enabled, the operating system determines the largest common path maximum transmission unit (PMTU)

value between servers and uses it as the unit size. A routing table entry is also created for each client network that attempts to connect to the server.

If you have a poorly performing Internet server that handles mainly remote traffic and the routing table increases to more than 1000 entries, disabling the PMTU discovery can decrease the size of the routing table, which may improve server efficiency. However, if a server handles mainly local traffic and only some remote traffic, disabling PMTU discovery can degrade bandwidth. Use the `netstat -r` command to display the contents of the routing table.

The default value is 1 (PMTU enabled). The *recommended value* is 0 (PMTU disabled).

Increasing the Number of Outgoing Connection Ports

When a TCP or UDP application creates an outgoing connection, the kernel dynamically allocates a nonreserved port number for each connection.

The kernel selects the port number from a range of values between `ipport_userreserved_min` (if you are running Tru64 UNIX Version 4.0E or higher) or 1024 (if you are running Version 4.0D or earlier), and `ipport_userreserved`.

Using the default attribute values, the range of outgoing ports begins at port 1024 and ends at port 5000, and the number of simultaneous outgoing connections is limited to 3976 (5000 minus 1024).

If you have a proxy server, caching server, gateway system, or firewall system, with a load of more than 4000 simultaneous connections, you can modify the value of the `ipport_userreserved` attribute. The default value is 5000, which is the minimum value. The *recommended value* is 65535, which is the maximum value. Do not specify a value that is greater than 65535 or lower than 5000.

Modifying Process Attributes

You may be able to improve Internet server performance by tuning the following process (`proc`) subsystem attributes:

- `maxusers` (*Increasing the Size of System Tables and Data Structures*)
- `max_proc_per_user` (*Increasing the Number of Processes per User*)
- `max_threads_per_user` (*Increasing the Number of Threads per User*)
- `max_per_proc_data_size` (*Increasing the User Process Data Segment Size Limits*)

- `max_per_proc_address_space` (*Increasing the User Process Address Space Limits*)

These attributes set limits on system resources. If your Internet server appears to be reaching the resource limits, you may want to increase the value of one or more of these attributes. However, increasing the value of these attributes will allow the system to consume more memory.

Increasing the Size of System Tables and Data Structures

System algorithms use the `maxusers` attribute to size various system data structures and system tables. Increasing the value of `maxusers` provides more system resources to processes. However, this will increase the amount of wired memory.

If your system experiences a lack of resources (for example, Out of processes, No more processes, or pid table is full messages) and you have sufficient memory, increase the value of the `maxusers` attribute.

To determine an appropriate value for the `maxusers` attribute, you can double the default value until you improve performance. For example, if you have up to 1 GB of memory, increase the value of the `maxusers` attribute to 512. If you have up to 2 GB, increase the value to 1024. If you have an Internet, Web, proxy, caching, firewall, or gateway server, increase the value of the `maxusers` attribute to 2048.

The default value varies from 16 to 2048, depending on the amount of physical memory in the system. It is not recommended that you increase the value to more than 2048.

System administrators can change the `maxusers` attribute with the following command:

```
# sysconfig -r proc maxusers=N
```

The value *N* is the desired new value. For Tru64 UNIX Version 5.0A or higher, this command triggers the automatic expansion of the `pid` table. The resizing of other system tables is not performed until you specify a new value for the `maxusers` attribute in the `/etc/sysconfigtab` file and reboot the system.

Increasing the Number of Processes per User

The `max_proc_per_user` attribute specifies the maximum number of processes that can be allocated at any one time to each user, except superuser.

If your system experiences a lack of processes, increase the value of this attribute. If you have a multiprocess Internet server (for example, a server running IPlanet, Apache, CERN, or Zeus), you also may want to increase the value of this attribute. Note that increasing its value increases the amount of wired memory.

The default value is 64. The *recommended value* is 2000. The value you choose must not be more than the maximum number of processes that can be started by your system. For Internet servers, these processes include CGI processes. If you specify a value of 0 (zero) for this attribute, there is no limit on the number of processes per user.

Increasing the Number of Threads per User

The `max_threads_per_user` attribute specifies the maximum number of threads that can be allocated at any one time to each user, except superuser.

If your system experiences a lack of threads, increase the value of this attribute. If you have a multithreaded Internet server (for example, a server running Netscape FastTrack or Netscape Enterprise), you also may want to increase the value this attribute.

The default value is 256. The *recommended value* is 4096. The value must not be more than the maximum number of threads that can be started by your system.

Increasing the User Process Data Segment Size Limits

The `max_per_proc_data_size` attribute specify the maximum limit of data segment sizes. Some large programs and large-memory processes may not run unless you increase the values of this attribute. Increase the limits if you receive an Out of process memory message.

The default value is 1073741824 (1 GB). The *recommended value* is 10737418240 (10 GB). If your system has more than 10 GB of memory, you can further increase this value.

Increasing the User Process Address Space Limits

The `max_per_proc_address_space` attribute specifies the maximum limit of user process address space (number of bytes of virtual memory).

Some large programs and large-memory processes may not run unless you increase the value of this attribute. However, increasing the address space limits will cause a small increase in memory consumption.

The default value is 1073741824 (1 GB) for systems running Tru64 UNIX Version 4.0G and earlier, and 4294967296 (4 GB) for systems running Tru64 UNIX Version 5.0 or higher.

The *recommended value* is 10737418240 (10 GB). If your system has more than 10 GB of memory, you can further increase this value.

Modifying Socket Attributes

You may be able to improve Internet server performance by tuning the following socket attributes:

- `somaxconn` (*Increasing the Maximum Number of Pending TCP Connections*)
- `sominconn` (*Increasing the Minimum Number of Pending TCP Connections*)
- `sbcompress_threshold` (*Enabling the mbuf Cluster Compression*)

Increasing the Maximum Number of Pending TCP Connections

The `somaxconn` attribute specifies the maximum number of pending TCP connections (the socket listen queue limit) for each server socket (for example, for the HTTP server socket). Pending TCP connections can be caused by lost packets in the Internet or denial of service attacks. Busy Internet servers often experience large numbers of pending connections. If the listen queue connection limit is too small, incoming connect requests may be dropped.

The default value is 1024. The *recommended value* is 65535, which is the maximum value. Do not specify a value that is higher than the maximum value because this can cause unpredictable behavior.

Increasing the Minimum Number of Pending TCP Connections

The `sominconn` attribute specifies the minimum number of pending TCP connections (backlog) for each server socket. The attribute controls the maximum number of SYN packets that the system can handle simultaneously before additional requests are discarded. Network

performance can degrade if a client saturates a socket listen queue with erroneous TCP SYN packets, which blocks other users from the queue.

The value of the `sominconn` attribute overrides the application-specific backlog value, which may be set too low for some server software. If you do not have your application source code, use the `sominconn` attribute to set a pending-connection limit that is appropriate for your application.

The default value is 0. The *recommended value* is 65535, which is the maximum value. It is recommended that the value of the `sominconn` attribute be the same as the value of the `somaxconn` attribute. See *Increasing the Maximum Number of Pending TCP Connections* for more information in the `somaxconn` attribute.

Enabling the mbuf Cluster Compression

The `sbcompress_threshold` attribute controls whether mbuf clusters are compressed at the socket layer. By default, mbuf clusters are not compressed, which can cause proxy servers and caching servers to consume all the available mbuf clusters. This problem is more likely to occur if you are using FDDI, instead of Ethernet. See *Tuning Tru64 UNIX for Internet Servers* for information about monitoring mbuf clustering.

To enable mbuf cluster compression, modify the `sbcompress_threshold` attribute and specify a value. Packets will be copied into the existing mbuf clusters if the packet size is less than this value.

The default value is 0 (mbuf compression is disabled). If you have a proxy server, caching server, gateway system, or firewall system, the *recommended value* is 600 bytes.

Modifying Virtual Memory Attributes

For systems running Tru64 UNIX Version 4.0G and earlier, you may be able to improve Internet server performance by tuning the following Virtual Memory (vm) subsystem attributes:

- `vm-mapentries` (*Increasing the Maximum Number of Memory-Mapped Files*)
- `vm-maxvas` (*Increasing the Maximum Amount of Valid Virtual Address Space*)

Increasing the Maximum Number of Memory-Mapped Files

The `vm-mapentries` attribute specifies the maximum number of memory-mapped files for a user process for Tru64 UNIX Version 4.0G and earlier.

The operating system limits the number of memory-mapped files that are available to each process. Each map entry describes one unique disjoint portion of a virtual address space.

The two primary types of Internet servers are multiprocess and multithreaded Internet servers. Because multithreaded Internet servers are more likely to use memory-mapped files, you may want to increase the maximum number of memory-mapped files if you have this type of system.

In addition, after several hours of use, Netscape Enterprise Server users may receive forbidden messages in response to Web page requests. The errors file may also contain a `URL could not load message`, where URL specifies the location of the requested page. If this occurs, the server may have the value of the `vm-mapentries` attribute set too low.

The default value is 200 map entries. If you have a busy very-large memory (VLM) system running multithreaded Internet server software, the *recommended value* is 20000. The `vm-mapentries` attribute affects all processes, so increasing its value will allow the system to consume more memory.

Increasing the Maximum Amount of Valid Virtual Address Space

The `vm-maxvas` attribute specifies the maximum amount of valid virtual address space for a process (that is, the sum of all the valid pages) for Tru64 UNIX Version 4.0G and earlier.

The default value is 1073741824 bytes (1 GB). If you have an Internet, Web, proxy, caching, firewall, or gateway server, the *recommended value* is 10737418240 (10 GB).

Troubleshooting

If performance is still deficient after applying the primary Internet server tuning recommendations, you may be able to improve performance by modifying additional kernel subsystem attributes, as discussed in the *Tuning Tru64 UNIX for Internet Servers* guide.

You can also use the Compaq Continuous Profiling Infrastructure (CPI, formerly known as DCPI) tool to obtain detailed information about system components that heavily utilize CPU cycles. CPI is offered as an Advanced Development Kit. Refer to the following site for more information:

<http://www.tru64unix.compaq.com/dcpi>

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

Compaq and the Compaq logo Registered in U.S. Patent and Trademark Office. Tru64 is a trademark of Compaq Information Technologies Group, L.P. in the United States and other countries. UNIX is a trademark of The Open Group in the United States and other countries. All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from Compaq required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.