

# Tru64 UNIX

---

## Improving Large-Memory System Performance

**March 2002**

**Product Version:** Tru64 UNIX Version 4.0G and higher

This document contains information that systems engineers need when improving large-memory system performance on the Tru64™ UNIX operating system.

---

© 2002 Compaq Information Technologies Group, L.P.

Compaq, the Compaq logo, AlphaServer, and TruCluster Registered in U.S. Patent and Trademark Office. Alpha, OpenVMS, and Tru64 are trademarks of Compaq Information Technologies Group, L.P. in the United States and other countries.

UNIX is a registered 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.

---

# Contents

## About This Document

### 1 Displaying and Modifying System Performance

|     |  |     |
|-----|--|-----|
| 1.1 | Displaying Memory Usage by Using the vmstat Command ....                       | 1-1 |
| 1.2 | Displaying Memory and Process Information by Using the ps Command .....        | 1-2 |
| 1.3 | Displaying Disk Usage by Using the iostat Command .....                        | 1-3 |
| 1.4 | Displaying Swap Space Usage by Using the swapon Command .....                  | 1-4 |
| 1.5 | Displaying AdvFS Buffer Cache Hit Rate by Using the advfsstat Command .....    | 1-4 |
| 1.6 | Displaying System Configuration Information and Analysis Using sys_check ..... | 1-6 |

### 2 Managing Kernel Subsystem Attributes

|       |   |     |
|-------|---|-----|
| 2.1   | Operating System Support for Attributes ..... | 2-1 |
| 2.2   | Displaying Attribute Values .....             | 2-2 |
| 2.3   | Modifying Attribute Values .....              | 2-3 |
| 2.3.1 | Modifying the Current Value .....             | 2-3 |
| 2.3.2 | Modifying the Permanent Value .....           | 2-4 |

### 3 Primary Tuning Recommendations

|         |  |     |
|---------|--|-----|
| 3.1     | Modifying Advanced File System Attributes .....  | 3-2 |
| 3.1.1   | Modifying the Size of the AdvFS Buffer Cache Size for Tru64 UNIX Version 4.0F through 5.0A ..... | 3-3 |
| 3.1.2   | Modifying AdvFS Cache Hash Chains for Tru64 UNIX Version 4.0F, 4.0G, 5.0 and 5.0A .....          | 3-4 |
| 3.1.3   | Modifying AdvFS Access Structures Allocated at Boot Time .....                                   | 3-5 |
| 3.2     | Modifying Process Attributes .....   | 3-6 |
| 3.2.1   | Increasing the Size of System Table Space and Data Structure Resources .....                     | 3-7 |
| 3.2.2   | Increasing Virtual Address Space Resources .....   | 3-8 |
| 3.2.2.1 | Increasing User Process Address Space .....  | 3-8 |

|         |   |      |
|---------|---|------|
| 3.2.2.2 | Increasing the Limit on File Mapping for Tru64 UNIX<br>Version 4.0G and 4.0F .....                                      | 3-9  |
| 3.2.3   | Increasing the Maximum Number of Processes per User ..  | 3-9  |
| 3.2.4   | Increasing the Maximum Number of Threads Per User ...   | 3-9  |
| 3.2.5   | Increasing the Maximum Sizes of a User Process Stack<br>Size .....  | 3-9  |
| 3.2.6   | Increasing the User Process Data Segment Size Limits ...  | 3-10 |
| 3.3     | Modifying Virtual File System Attributes .....  | 3-10 |
| 3.3.1   | Increasing the Number of Open Files .....   | 3-11 |
| 3.3.2   | Specifying the Number of Pages in the Traditional Buffer<br>Cache .....   | 3-11 |
| 3.3.3   | Reducing the Size of the Traditional Buffer Cache .....   | 3-11 |
| 3.3.4   | Increasing the Size of the Namei Cache .....  | 3-12 |
| 3.3.5   | Increasing the Size of the Namei Cache Hash Table .....   | 3-12 |
| 3.4     | Modifying Interprocess Communication Attributes .....   | 3-12 |
| 3.4.1   | Increasing the System V Shared Memory Region .....  | 3-14 |
| 3.4.2   | Increasing the Shared Memory Regions Attached to a<br>Process .....   | 3-14 |
| 3.4.3   | Increasing the Shared Memory Regions That Can be Used<br>on the System at One Time .....                                | 3-14 |
| 3.4.4   | Increasing the Shared Page Table Limit .....  | 3-15 |
| 3.4.5   | Increasing System V Semaphores That Can be Used on<br>the System at One Time .....                                      | 3-15 |
| 3.4.6   | Increasing System V Semaphores That Can be Used by a<br>Single Process at One Time .....                                | 3-15 |
| 3.4.7   | Increasing Operations That Can Be Outstanding on a<br>Single System V Semaphore at One Time .....                       | 3-15 |
| 3.4.8   | Increasing the Number of Undo Operations That Can be<br>Outstanding on a Single System V Semaphore at One<br>Time ..... | 3-16 |
| 3.5     | Modifying Virtual Memory Attributes .....   | 3-16 |
| 3.5.1   | Reducing the Amount of Memory Allocated to the UBC ...  | 3-17 |
| 3.5.2   | Specifying Sequential File Caching .....  | 3-18 |
| 3.5.3   | Increasing the Paging Threshold .....   | 3-19 |
| 3.5.4   | Tuning Swapping Threshold .....   | 3-20 |
| 3.5.5   | Increasing the Rate of Modified Page Prewriting .....   | 3-20 |

---

## About This Document

This document describes how to tune the Compaq Tru64™ UNIX operating system to improve large-memory system performance. Large-memory systems (greater than 2 GB) often run memory-intensive applications, such as database programs, that require significant system resources. This document describes how to configure swap space, increase system resources for large-memory systems, and improve system performance by modifying some kernel subsystem attributes.

### Audience

This document is intended for system administrators who are responsible for managing a Tru64 UNIX operating system running more than 2 GB of physical memory. Administrators should have an in-depth knowledge of their applications and users, in addition to operating system concepts, commands, and utilities.

It is also important for the administrator to understand how to access and modify kernel subsystem attributes and understand paging, swapping, and file-system caching for their version of the operating system. Understanding these topics is crucial to successfully tuning a system for better performance.

### Organization

This document is organized as follows:

|                  |  |
|------------------|--|
| <i>Chapter 1</i> | Provides information on how to display and modify system performance.            |
| <i>Chapter 2</i> | Provides information on how to manage kernel subsystem attributes.               |
| <i>Chapter 3</i> | Provides primary tuning recommendations and examples on how to tune your system. |

### Related Documentation

The following documentation supplements information in this document:

#### Best Practices

For information on improving the performance of Internet servers, including Web servers, ftp servers, mail servers and relays, proxy servers, caching

servers, gateway systems, and firewall systems, refer to Tuning Tru64 UNIX for Internet Servers at the following URL:

[http://www.tru64unix.compaq.com/docs/best\\_practices/BP\\_INTUNING/TITLE.HTM](http://www.tru64unix.compaq.com/docs/best_practices/BP_INTUNING/TITLE.HTM)

## Manuals

The following Tru64 UNIX operating system manuals provide important information that supplements the information in this document:

- *System Configuration and Tuning* contains detailed information on configuring and tuning high-performance and high availability systems.
- *System Administration* describes how to configure, use, and maintain the Tru64 UNIX operating system.

You can access these manuals on line for your version of the operating system at the following URL:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

## Reference Pages

The following Tru64 UNIX reference pages contain descriptions of the subsystem attributes:

- `sys_attrs(5)`
- `sys_attrs_advfs(5)`
- `sys_attrs_ipc(5)`
- `sys_attrs_proc(5)`
- `sys_attrs_vfs(5)`
- `sys_attrs_vm(5)`

You can access these reference pages on line for your version of the operating system at the following URL:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

## Reader's Comments

Compaq welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/X66
- Internet electronic mail: [readers\\_comment@zk3.dec.com](mailto:readers_comment@zk3.dec.com)

A Reader's Comment form is located on your system in the following location:

`/usr/doc/readers_comment.txt`

Please include the following information along with your comments:

- The full title of the book and the order number. (The order number is printed on the title page of this book and on its back cover.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate Compaq technical support office. Information provided with the software media explains how to send problem reports to Compaq.



---

## Displaying and Modifying System Performance

This chapter describes the commands that you can use to display and modify system performance statistics. Gathering performance information about the current configuration can help determine which tuning recommendations to apply. Save the output of the monitoring commands. After tuning, you can use the information to determine if you have improved system performance.

The following commands let you display and modify system performance:

- `vmstat` — displays memory usage statistics (Section 1.1)
- `ps` — displays memory and process information (Section 1.2)
- `iostat` — displays disk usage statistics (Section 1.3)
- `swapon` — displays the swap usage statistics for your system (Section 1.4)
- `advfsstat` — displays the Advanced File System (AdvFS) buffer cache hit rate (Section 1.5)
- `sys_check` — displays system configuration information and analysis (Section 1.6)

### 1.1 Displaying Memory Usage by Using the `vmstat` Command

Use the `vmstat` command to display the virtual memory, process, and CPU usage statistics for a specified time interval. The first line of the output is for all time since a reboot, and each subsequent report is for the last interval. Use the output to check the number of free pages (`free`) and to determine if the system is paging (`pout`).

An example of the `vmstat` command is as follows; output is provided in one-second intervals:

```
# vmstat 1
Virtual Memory Statistics: (pagesize = 8192)
procs  memory      pages      intr      cpu
r  w  u  act  free wire fault  cow zero react pin pout  in  sy  cs  us  sy  id
7 526 59 80K 758 45K 402M 94M 132M 1M 74M 139K 757 42K 1K 38 14 48
7 526 59 81K 278 45K 939 15 896 0 11 0 824 2K 1K 85 11 4
6 528 59 81K 285 45K 595 67 411 0 10 0 983 5K 2K 81 17 2
7 526 59 81K 353 45K 560 31 446 0 17 0 781 2K 1K 87 10 3
7 526 59 81K 353 45K 406 0 406 0 0 0 1K 4K 2K 85 13 2
7 527 59 81K 288 45K 406 0 406 0 0 0 1K 7K 4K 81 18 1
```

```

9 524 59 81K 350 45K 640 72 420 0 13 0 999 3K 2K 85 13 2
.
.
.

```

The values in the memory fields are specified in 8-KB pages. Check the size of the free page list (*free*). Compare the number of free pages to the values for the active pages (*act*) and the wired pages (*wire*). The sum of the free, active, and wired pages should be close to the amount of physical memory in your system. Although the value for *free* should be small, if the value is consistently small (less than 128 pages) and accompanied by excessive paging and swapping, you may have a physical memory shortage. Pages that have been paged out to disk (*pout*), should be very low (in single digits) or zero. Pages out are always disk I/O.

For more information on *vmstat* command, see the *System Configuration and Tuning* manual or the *vmstat(1)* for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

## 1.2 Displaying Memory and Process Information by Using the *ps* Command

Use the *ps* command to display the current status of the system processes. You can use the output to determine the current running processes (including users), their state, and how they utilize system memory. The *-u* option lists processes in order of decreasing CPU usage, so you can identify which processes are using the most CPU time. From the output of the *ps* command, you can determine which processes are consuming most of your system's CPU time and memory resources, and whether processes are swapped out (*w*).

An example of the *ps* command is as follows:

```

# /usr/ucb/ps aux
USER  PID  %CPU %MEM  VSZ   RSS TTY  S   STARTED      TIME COMMAND
chen  2225  5.0  0.3  1.35M 256K p9  U   13:24:58  0:00.36 cp /vmunix /tmp
root  2236  3.0  0.5  1.59M 456K p9  R   + 13:33:21  0:00.08 ps aux
sorn  2226  1.0  0.6  2.75M 552K p9  S   + 13:25:01  0:00.05 vi met.ps
root   347  1.0  4.0  9.58M 3.72 ??  S   Nov 07  01:26:44 /usr/bin/X11/X -a
root  1905  1.0  1.1  6.10M 1.01 ??  R   16:55:16  0:24.79 /usr/bin/X11/dxpa
mat   2228  0.0  0.5  1.82M 504K p5  S   + 13:25:03  0:00.02 more
jayh  1016  0.0  0.0  6.70M  0K ??  IW   Nov 19  0:00.10 gcc
jayh   321  0.0  0.0  80.2M  0K ??  IW   Nov 18  0:00.91 mozilla
.
.
.

```

The process virtual address size field (*VSZ*) specifies the total amount of virtual memory allocated to the process (in bytes). Check the processes that are swapped out (*w*). If processes are continually being swapped out, this could indicate a lack of memory resources.

A multithreaded application on a multiprocessing system may show CPU usage that is more than 100 percent. The BSD compatible option `m` (these options are not prefixed with a dash (-) character) prints all threads in a task, if the task has more than one.

The `ps` command provides only a snapshot of the system; by the time the command finishes executing, the system state has probably changed. In addition, one of the first lines of the command may refer to the `ps` command itself. The `aux` option sorts procs by physical memory consumption.

For more information on the `ps` command, see the *System Configuration and Tuning* manual or `ps(1)` for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

### 1.3 Displaying Disk Usage by Using the `iostat` Command

For optimum system performance, evenly distribute disk I/O across disks and, ideally, across buses. Use the `iostat` command to determine which disks are being used the most. The `iostat` command displays disk I/O statistics for disks, as well as terminal and CPU statistics.

An example of the `iostat` command is as follows; output is provided in one-second intervals:

```
# /usr/ucb/iostat 1
tty      floppy0      dsk0      dsk1      cdrom0      cpu
tin tout   bps tps   bps tps   bps tps   bps tps   us ni sy id
1   73     0  0    23  2    37  3     0  0     5  0 17 79
0   58     0  0    47  5   204 25     0  0     8  0 14 77
0   58     0  0     8  1    62  1     0  0    27  0 27 46
```

For the output in the system column (`cpu`), the `iostat` command displays the percentage of time the CPU spent in:

- User state running processes at either their default priority or preferred priority (`us`)
- User mode running processes at a less favored priority (`ni`)
- System mode (`sy`)
- Idle mode (`id`)

This information lets you determine how disk I/O is affecting the CPU.

For more information on the `iostat` command, see the *System Configuration and Tuning* manual or `iostat(1)` for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

## 1.4 Displaying Swap Space Usage by Using the swapon Command

Use the `swapon -s` command to display your swap device configuration. For each swap partition, the command displays the total amount of allocated swap space, the amount of swap space that is being used, and the amount of free swap space. This information can help you determine how your swap space is being utilized.

An example of the `swapon` command is as follows:

```
# /usr/sbin/swapon -s
Swap partition /dev/disk/dsk1b (default swap):
  Allocated space:      16384 pages (128MB)
  In-use space:         10452 pages ( 63%)
  Free space:           5932 pages ( 36%)

Swap partition /dev/disk/dsk4c:
  Allocated space:      128178 pages (1001MB)
  In-use space:         10242 pages (  7%)
  Free space:           117936 pages ( 92%)

Total swap allocation:

  Allocated space:      144562 pages (1.10GB)
  Reserved space:       34253 pages ( 23%)
  In-use space:         20694 pages ( 14%)
  Available space:      110309 pages ( 76%)
```

If you have less than 50 percent total available swap space, consider adding swap space. You can optimize performance by having all your swap space configured when you boot the system, instead of adding swap space while the system is running.

For more information about the `swapon` command, see the *System Configuration and Tuning* manual or `swapon(8)` for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

## 1.5 Displaying AdvFS Buffer Cache Hit Rate by Using the advfsstat Command

Use the `advfsstat -b` command to check the Advanced File System (AdvFS) buffer cache hit rate for an AdvFS domain. The output shows the number of AdvFS read and write requests, and how often a page was found in the AdvFS buffer cache or the UBC for an AdvFS domain.

Use the `advfsstat -b -s` command to display statistics since boot time for a domain. For example:

```
# advfsstat -b -s staff_domain
pin      ref      unpin-type      misc      cons
cnt hit hitw read cnt hit hitw lazy blk cln log ra ubc un abrt
```

```
1M 1M 462 0 2M 2M 1974 1M 1941 1976 64K 68K 66K 0 0
```

To determine if your AdvFS domain's buffer cache activity is performing well for your workload, calculate the overall domain cache hit rate by multiplying the overall hit ratio by 100.

To obtain the overall AdvFS buffer cache hit ratio for a domain, use the following formula:

```
(pin hit + ref hit) / (pin cnt + ref cnt)
```

*For Tru64 UNIX Versions 4.0F, 4.0G, 5.0 and 5.0A:* Recent AdvFS read and write requests are stored in buffers. When an application requests a page, AdvFS checks the AdvFS buffer cache for the page, and then checks the Unified Buffer Cache (UBC). If it finds the page, it reuses the buffer. If the page is not found in the cache, the page is obtained from disk. Reusing buffers reduces disk I/O operations and can improve file system performance. As AdvFS requires pages for newly referenced data, it recycles the buffers in the AdvFS buffer cache.

In some cases, you may be able to increase the AdvFS buffer cache hit rate by increasing the size of the cache. In addition, you may be able to improve cache search times by increasing the number of caches. In addition, you may be able to improve cache search times by increasing the number of cache hash chains. However, if your configuration does not reuse AdvFS data a lot, you may want to decrease the size of the AdvFS buffer cache. See Section 3.1.1 and Section 3.1.2 for information.

*For Tru64 UNIX Versions 5.1 and higher:* Recent AdvFS read and write requests are stored in the UBC. When an application requests a page, AdvFS checks the UBC for the page. If the page is found in the UBC, AdvFS reuses the cache page. If the page is not found in the cache, the page is obtained from disk and stored in a new UBC page. Reusing buffers reduces disk I/O operations and can improve file system performance. The UBC decides when to recycle existing cache pages.

Adjust the UBC parameters to tune the caching behavior for AdvFS. AdvFS now uses only the UBC and no longer manages its own buffer cache. Changing UBC parameters adjusts the caching behavior on a system-wide basis. The `advfsstat` command displays AdvFS caching usage on a per domain basis. You may use this `advfsstat` information as part of your assessment of the system's caching behavior. See Section 3.5 for more information on tuning UBC.

For more information about modifying cache hash chain lengths, see Section 3.1.2.

For more information on the `advfsstat` command, see the *System Configuration and Tuning* manual or `advfsstat(8)` for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

## 1.6 Displaying System Configuration Information and Analysis Using `sys_check`

The `sys_check` utility performs an analysis of operating system parameters and attributes that tune the performance of your system. The report generated by `sys_check` provides warnings if it detects problems with any current settings. Use `sys_check` utility in conjunction with event management and system monitoring tools to provide a complete overview and control of system status.

Run the `sys_check` utility with the `-perf` option, and consider applying its tuning recommendations. The utility may take 5 to 15 minutes to complete.

For more information, see the *System Configuration and Tuning* manual or `sys_check(8)` for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

---

## Managing Kernel Subsystem Attributes

The Tru64 UNIX operating system includes various subsystems that define or extend the kernel. This chapter describes how 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.

For introductory information about kernel subsystem attributes, see `sys_attrs(5)` for your version of the operating system.

The following list describes how to manage kernel subsystem attributes:

- Determine the operating system support for an attribute (Section 2.1)
- Display the attribute values (Section 2.2)
- Modify the attribute values (Section 2.3)

### 2.1 Operating System Support for Attributes

Before you start managing the kernel subsystem attributes you need to determine if your version of the operating system will support specific attributes.

---

#### Note

Tru64 UNIX Version 4.0F and 4.0G restrict the methods that you can use to modify and display attributes, or they require operating system patches to use the attributes.

---

To determine if your version of the operating system supports a particular kernel subsystem attribute, use one of the following methods:

Use the `sysconfig -q subsystem [attribute]` command.

If you do not specify an attribute, the system displays all the subsystem attributes that can be modified with the `sysconfig` or `sysconfigdb` command. If the subsystem is not configured, the operating system displays a message similar to the following:

```
framework error: subsystem 'inet' not found
```

If you specify an attribute, only the information specific to that attribute is displayed. For example:

```
# sysconfig -q inet tcbhashsize
inet:
tcbhashsize = 32
```

If the attribute is not supported or if it cannot be accessed by using `sysconfig`, the operating system displays a message similar to the following:

```
inet:
tcbhashsize = unknown attribute
```

For more information, see `sysconfig(8)` and `dbx(1)` for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

## 2.2 Displaying Attribute Values

There are various methods that you can use to display the current value of a kernel subsystem attribute and other descriptive information. Use any of the following methods to display attribute values:

- The Kernel Tuner (`dxkerneltuner`) graphical user interface (GUI) displays permanent, current (run-time), minimum, and maximum values of attributes. Access the GUI through the Common Desktop Environment (CDE) Application Manager window; select the `System_Admin` icon, and then select the `MonitoringTuning` icon. You can then choose the subsystem whose attributes you want to display.
- The `sysconfig -q subsystem [attribute]` command displays the current (run-time) value of the specified attribute or, if an attribute is not specified, all the attributes for the specified subsystem:

```
sysconfig -q subsystem [attribute]
```

For example:

```
# sysconfig -q vm ubc_maxpercent
vm:
ubc_maxpercent = 100
```

- The `sysconfig -Q subsystem [attribute]` command displays the maximum and minimum values of the attributes for the specified subsystem. If you specify a particular attribute, the system displays information only for that attribute.

The command output also includes information about the operations that you can perform on the attribute. The following lists various operations:

- C — The attribute can be modified when the subsystem is initially loaded; that is, the attribute supports boot time, permanent modifications.

- R — The attribute can be tuned at run time; that is, you can modify the value that the system is currently using.
- Q — The attribute's current value can be displayed (queried).

For example:

```
# sysconfig -Q vfs bufcache
vfs:
bufcache -      type=INT op=CQ min_val=0 max_val=50
```

The output of the previous command shows that the minimum value of the `bufcache` attribute is 0 and the maximum value is 50. The output also shows that you cannot modify the current (run-time) value.

For more information, see `dxkerneltuner(8X)`, `sysconfig(8)`, and `dbx(1)` for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

## 2.3 Modifying Attribute Values

There are various methods you can use to modify attribute values. The method you use depends on the version of the operating system you are running, as specified in Section 2.1, and whether you want to modify the current (run-time) value of an attribute or modify an attribute's permanent value. Schedule system tuning before a planned shutdown to maintain availability. You must be root to modify attribute values.

The following sections describe how to modify the current and permanent values.

### 2.3.1 Modifying the Current Value

In some cases, you can modify the current (run-time) value of an attribute. This allows you to determine if modifying an attribute will improve your system performance without rebooting the system. Not all attributes can be tuned at run time, and the temporary modifications are lost when you reboot the system. Use the `sysconfig -Q` command to determine whether an attribute can be tuned at run time.

To modify an attribute's current (run-time) value, use one of the following methods:

- The Kernel Tuner (`dxkerneltuner`) GUI, if the attribute supports this operation. 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.

- The `sysconfig -r` command, if the attribute supports this operation. Use the following command syntax:

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

For example:

```
# sysconfig -r inet tcp_keepinit=30
tcp_keepinit: reconfigured
```

For more information, see `dxkerneltuner(8X)`, `sysconfig(8)`, and `dbx(1)` for your version of the operating system.

## 2.3.2 Modifying the Permanent Value

To modify an attribute's permanent (boot-time) value, the `sysconfigtab` file must contain the subsystem name, the attribute name, and the value of the attribute. Do not manually modify the `sysconfigtab` file. To make these modifications, use one of the following methods:

- The Kernel Tuner (`dxkerneltuner`) 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 `Boot Time Value` field.
- The `sysconfigdb` command. Use the following command syntax:

```
sysconfigdb -a -f stanza_file subsystem
```

The *stanza\_file* is a specially formatted file that contains the name of the subsystem and a list of attributes and their values. This file is merged into the `sysconfigtab` file.

To use the new attribute value, you must invoke the `sysconfig -r` command if the attribute can be tuned at run time, or reboot the system.

In addition, you can use the `dbx patch` command to modify the value of a variable, as well as the on-disk `/vmunix` image value.

See `dxkerneltuner(8X)`, `sysconfig(8)`, `sysconfigdb(8)`, `stanza(4)`, and the *System Administration* manual for your version of the operating system for information about modifying the system configuration file:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

---

## Primary Tuning Recommendations

You may be able to improve performance by tuning the default values of kernel attributes that control system resource allocation and subsystem behavior, including the virtual memory, file-system, and network subsystems. In some cases, the default values are inadequate for large-memory systems, systems running applications that require extensive resources, or highly specialized systems, such as Internet servers.

To determine which kernel attributes you need to modify, you must understand how your applications use memory and CPU resources, perform disk and file system I/O, and perform network I/O.

There are many kernel subsystem attributes that affect large-memory system performance. This chapter offers primary tuning recommendations for some of the attributes for the following subsystems:

- Advanced File System (AdvFS) (Section 3.1)
- Process (Section 3.2)
- Virtual file system (Section 3.3)
- Interprocess communication (Section 3.4)
- Virtual memory (Section 3.5)

---

### Note

---

Some kernel subsystem attributes enable you to modify their value and apply the value to a running system. Other attributes require you to reboot the system to use a new value. See Section 2.2 to determine if an attribute can be tuned at run time.

---

Before you reboot the system to apply new values to many kernel subsystem attributes, you may want to tune the system immediately before scheduled system down time to decrease the impact on availability. We recommend that you apply and check the tuning recommendations on a test or staging system before deploying the changes on a production system.

The primary tuning recommendations provide the best performance improvement for most large-memory configurations. If performance is still deficient after applying these recommendations, you may be able to

improve performance by modifying additional kernel subsystem attributes, as discussed in the *System Configuration and Tuning* manual.

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. See the following Web site for more information:

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

### 3.1 Modifying Advanced File System Attributes

This section describes how to improve performance by modifying the following Tru64 UNIX AdvFS kernel subsystem attributes for your version of the Tru64 UNIX operating system:

| Attribute              | Description  | 4.0F | 4.0G | 5.0 | 5.0A | 5.1 | 5.1A |
|------------------------|--|------|------|-----|------|-----|------|
| AdvfsCacheMax-Percent  | Specifies the size of the AdvFS buffer cache (Section 3.1.1).  | Y    | Y    | Y   | Y    | N   | N    |
| AdvfsCacheHashSize     | Specifies the number of hash chains on the table (Section 3.1.2).  | Y    | Y    | Y   | Y    | N   | N    |
| AdvfsMinFreeAccess     | The minimum number of AdvFS access structures that can be on the access structure free list at one time (Section 3.1.3). | N    | N    | Y   | Y    | Y   | Y    |
| AdvfsAccessMax-Percent | The maximum percentage of the malloc pool that can be allocated for access structures (Section 3.1.3).                   | Y    | Y    | Y   | Y    | Y   | Y    |
| AdvfsPreallocAccess    | The number of access structures allocated on the AdvFS access structure free list at system start-up (Section 3.1.3).    | Y    | Y    | N   | N    | N   | N    |

#### Note

If you are running Tru64 UNIX Version 4.0F or 4.0G the kernel subsystem attributes are displayed using dashes ( - ); Tru64 UNIX Version 5.0 and higher the attributes are displayed using underscores ( \_ ).

For more information about the `advfs` kernel subsystems, see `sys_attrs_advfs(5)` for your version of the operating system.

---

**Note**

---

For Tru64 UNIX Version 5.1 and higher, the AdvFS specific buffer cache was integrated into the Unified Buffer Cache (UBC). Tune AdvFS caching by modifying the UBC parameters. For more information, see Section 1.5.

---

### 3.1.1 Modifying the Size of the AdvFS Buffer Cache Size for Tru64 UNIX Version 4.0F through 5.0A

The `advfs` subsystem attribute `AdvfsCacheMaxPercent` specifies the size of the AdvFS buffer cache. The default value is 7 percent of physical memory. The minimum value is 1 percent; the maximum value is 30 percent.

If your applications reuse data, you can maximize the number of AdvFS buffer cache hits by making the AdvFS buffer cache large enough to hold pages until they are reused. However, increasing the size of the AdvFS buffer cache consumes memory and may cause paging. If your applications do not reuse I/O, you do not need a large cache size. On very large-memory systems, the AdvFS buffer cache may be too large and may waste memory. You may want to free memory by decreasing the size of the cache in these cases.

To determine if you need to modify the size of the AdvFS buffer cache, see Section 1.5 for more information.

Use the following table to determine if you should modify the size of the AdvFS buffer cache. These recommendations apply only if AdvFS is configured on your system.

| Issues   | Tuning Recommendations  |
|--|---|
| <b>AdvFS is configured on your system, but you are not using AdvFS.</b>  | Set the value of the <code>AdvfsCacheMaxPercent</code> attribute to 1 to minimize the memory allocation to the AdvFS buffer cache.  |
| <b>You are running applications that perform their own buffer caching.</b>   | Decrease the value of the <code>AdvfsCacheMaxPercent</code> attribute by 1 or 2. Then, use the <code>advfsstat -b -s</code> command to check the hit rate. Continue to decrease only if the hit rate is above 80 percent.   |
| <b>You have a very-large memory system, with a moderate AdvFS load and an AdvFS buffer cache hit rate of more than 80 percent.</b> | An overly large buffer cache may be wasting memory. Consider decreasing the size of the AdvFS buffer cache. Reduce the value of <code>AdvfsCacheMaxPercent</code> by 1 or 2. Then, use the <code>advfsstat -b -s</code> command to check the hit rate. Continue to decrease only if the hit rate is above 80 percent. |

| Issues  | Tuning Recommendations   |
|---|--|
| <b>The system has an AdvFS buffer cache hit rate of more than 80 percent.</b> | Increase the available memory to cache AdvFS pages. To do this, increase the value of the <code>AdvfsCacheMaxPercent</code> attribute by 1 or 2. Continue to increase the value of <code>AdvfsCacheMaxPercent</code> only if you continue to improve the hit rate, without causing the system to page.   |
| <b>The system has a cache hit rate of less than 80 percent.</b>               | <p>This may indicate one of the following conditions:</p> <ul style="list-style-type: none"> <li>• The size of the AdvFS cache is insufficient for the load and pages are being recycled (used for a different AdvFS page) from the cache too quickly. In this case, you may want to increase the value of the <code>AdvfsCacheMaxPercent</code> attribute by 1 or 2.</li> <li>• <i>For Tru64 UNIX Version 4.0F through Version 5.0A:</i> Applications are mainly writing to new files and are not reusing the cached AdvFS pages. In some cases, increasing the value of the <code>AdvfsCacheMaxPercent</code> attribute by 1 or 2 may reduce cache recycling and provide better AdvFS I/O consolidation.</li> </ul> <p>Continue to increase <code>AdvfsCacheMaxPercent</code> only if you continue to increase the hit rate, without causing the system to page.</p> |

**Note**

For Tru64 UNIX Version 5.1 and higher, the AdvFS specific buffer cache was integrated into the Unified Buffer Cache (UBC). Tune AdvFS caching by modifying the UBC parameters. For more information, see Section 1.5.

### 3.1.2 Modifying AdvFS Cache Hash Chains for Tru64 UNIX Version 4.0F, 4.0G, 5.0 and 5.0A

The section is applies to Tru64 UNIX Versions 4.0F, 4.0G, 5.0 and 5.0A.

If you are running Tru64 UNIX Versions 5.1 or higher this parameter is obsolete. AdvFS uses the UBC for its caching and no longer maintains its own buffer caching. Tune the UBC parameters to adjust caching requirements for AdvFS.

The `advfs` subsystem attribute `AdvfsCacheHashSize` specifies the number of hash chains on the table. The default value is 8192 or 10 percent of the number of buffers in the AdvFS buffer cache (rounded up to the next power

of 2), whichever is the smallest value. The minimum value is 1024; the maximum value is either 65,536 or the size of the AdvFS buffer cache, whichever is the smallest value.

Make sure that your AdvFS buffer cache is the correct size before tuning hash chains see (Section 3.1.1).

To determine the number of buffers on each hash chain, follow these steps:

1. Determine the total number of AdvFS buffers. Use one of the following methods:
  - Examine the AdvFS start up message that displays the number of buffers. At boot time, the system allocates a number of AdvFS buffers and displays a message similar to the following:  

```
ADVFS: using 1611 buffers containing 12.58 megabytes of memory
```
  - Determine the number of 8-KB pages that can be obtained from the memory allocated to the AdvFS buffer cache. To do this, divide the value of the `AdvfsCacheMaxPercent` attribute (as a percentage of total physical memory) by 8192.
2. Divide the number of buffers by the number of hash chains, which is specified by the `AdvfsCacheHashSize` attribute.

Increase the number of hash chains only if there are more than 10 buffers on each hash chain. Set the attribute to 10 percent of the total number of AdvFS buffers allocated at boot time, up to a maximum of 65,536. In addition, if the size of your AdvFS buffer cache is less than 640 MB, you do not have to increase the number of hash chains.

---

**Note**

For Tru64 UNIX Version 5.1 and higher, the AdvFS specific buffer cache was integrated into the Unified Buffer Cache (UBC). Tune AdvFS caching by modifying the UBC parameters. For more information, see Section 1.5.

---

### 3.1.3 Modifying AdvFS Access Structures Allocated at Boot Time

The AdvFS subsystem attribute `AdvfsAccessMaxPercent` attribute controls the maximum percentage of pageable memory that can be allocated for AdvFS access structures. Caching open file information can enhance AdvFS performance if the open files are later reused. The minimum value is 5 percent; the maximum value is 95 percent.

The following table describes tuning recommendations for modifying AdvFS access structures allocated at boot time:

| Issue   | Tuning Recommendation  |
|---|--|
| <b>The system extensively reuses AdvFS files.</b> | <p>Consider increasing the memory allocated for the AdvFS access structure. However, this will consume memory and may cause paging. Use the <code>vmstat</code> command to make sure that you do not cause paging.</p> <ul style="list-style-type: none"> <li>For <i>Tru64 UNIX Version 5.0 through 5.1A</i>: At boot time and when the number of access structures on the free list is less than the value of the <code>AdvfsMinFreeAccess</code> attribute, AdvFS allocates additional access structures, until the number of access structures on the free list is twice the value of the <code>AdvfsMinFreeAccess</code> attribute or the value of the <code>AdvfsAccessMaxPercent</code> attribute, whichever is smaller. You can modify the attributes without rebooting the system.</li> <li>For <i>Tru64 UNIX Version 4.0F and 4.0G</i>: The <code>advfs</code> subsystem attribute <code>AdvfsPreallocAccess</code> specifies the number of AdvFS access structures that the system allocates at startup time. You must reboot the system to modify the attribute.</li> </ul> |

**Note**

For Tru64 UNIX Version 5.1 and higher, the AdvFS specific buffer cache was integrated into the Unified Buffer Cache (UBC). Tune AdvFS caching by modifying the UBC parameters. For more information, see Section 1.5.

## 3.2 Modifying Process Attributes

You can improve system performance by modifying the following Process (`proc`) kernel subsystem attributes for your version of the Tru64 UNIX operating system:

| Attribute             | Description  | 4.0F | 4.0G | 5.0 | 5.0A | 5.1 | 5.1A |
|-----------------------|--|------|------|-----|------|-----|------|
| <code>maxusers</code> | Specifies the number of simultaneous users that a system can support without straining system resources (Section 3.2.1). | Y    | Y    | Y   | Y    | Y   | Y    |

| Attribute   | Description  | 4.0F | 4.0G | 5.0 | 5.0A | 5.1 | 5.1A |
|---|--|------|------|-----|------|-----|------|
| max_per_proc_address_space and per_proc_address_space | Specifies the maximum amount, in bytes, of user process address space (Section 3.2.2).                                   | Y    | Y    | Y   | Y    | Y   | Y    |
| max_proc_per_user                                     | Specifies the maximum number of processes (tasks) that a user can create. The superuser is not affected (Section 3.2.3). | Y    | Y    | Y   | Y    | Y   | Y    |
| max_threads_per_user                                  | Specifies the maximum limit of threads a user can create. The superuser is not affected (Section 3.2.4).                 | Y    | Y    | Y   | Y    | Y   | Y    |
| max_proc_per_stack_size and per_proc_stack_size       | Specifies maximum size, in bytes, of a user process stack (Section 3.2.5).   | Y    | Y    | Y   | Y    | Y   | Y    |
| max_per_proc_data_size and per_proc_data_size         | Specifies current maximum size, in bytes, of a data segment for each process (Section 3.2.6).                            | Y    | Y    | Y   | Y    | Y   | Y    |

**Note**

If you are running Tru64 UNIX Version 4.0F or 4.0G the kernel subsystem attributes are displayed using dashes ( - ); if you are running Tru64 UNIX Version 5.0 and higher the attributes are displayed using underscores ( \_ ).

For more information about `proc` subsystem attributes see `sys_attrs_proc(5)` for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

### 3.2.1 Increasing the Size of System Table Space and Data Structure Resources

System algorithms use the `proc` subsystem attribute `maxusers` to size various system data structures and system tables, such as the system process table, which determines how many active processes can be running at one time. Increasing the value of the `maxusers` attribute 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, 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
```

*N* is the desired new value. For Tru64 UNIX Version 5.0A and 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.

## 3.2.2 Increasing Virtual Address Space Resources

The system allocates resources for virtual address space. You must check your virtual address space resource allocation, and increase it if necessary.

The following sections describe how to increase address space resources.

### 3.2.2.1 Increasing User Process Address Space

The `proc` subsystem attributes `per_proc_address_space` and `max_per_proc_address_space` specify the maximum limit of user process address space. Some large programs and large-memory processes may not run unless you increase the value of this attribute.

*For Tru64 UNIX Version 5.0 and higher:* The default value for `per_proc_address_space` and `max_per_proc_address_space` is 4,294,967,296 bytes (4 GB). The *recommended value* is 10,737,418,240 bytes (10 GB). If your system has more than 10 GB of memory, you can further increase this value.

*For Tru64 UNIX Version 4.0G and 4.0F:* The default value for `per-proc-address-space` and `max-per-proc-address-space` is 1 GB. If you increase the user process address space, you must also increase the default value of the `vm` subsystem attribute `vm-maxvas`, which specifies the maximum amount of valid virtual address space for a process. However,

increasing the address space limits will cause a small increase in memory consumption.

### 3.2.2.2 Increasing the Limit on File Mapping for Tru64 UNIX Version 4.0G and 4.0F

If you are running on a large-memory system or Internet server, you may want to increase the limit on file mapping. The `vm` subsystem attribute `vm-mapentries` specifies the maximum number of memory-mapped files in a user address. Each map entry describes one unique disjoint portion of a virtual address space.

The default value is 200.

### 3.2.3 Increasing the Maximum Number of Processes per User

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

If your system experiences a lack of processes, you may want to increase the maximum number of processes, or you can increase the value of the `maxusers` attribute. You can also specify a value for the `max_proc_per_user` attribute that is equal to or greater than the maximum number of processes that will be running on the system at one time.

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. If the value of the `max_proc_per_user` attribute is 0 (zero), there is no limit on processes; this is not recommended.

### 3.2.4 Increasing the Maximum Number of Threads Per User

The `proc` subsystem attribute `max_threads_per_user` 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.

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.

### 3.2.5 Increasing the Maximum Sizes of a User Process Stack Size

The `proc` subsystem attributes `per_proc_stack_size` and `max_per_proc_stack_size` specify the maximum sizes of a user process stack. Some large programs and large-memory processes may not run unless you increase the default value and maximum sizes. If you are running a large

program or a large-memory process, or if you receive a `cannot grow stack` message, increase the default and maximum sizes of a user process stack.

The default value of the `per_proc_stack_size` is 8,388,608 bytes (8 MB). The default value of the `max_per_proc_stack_size` attribute is 33,554,432 bytes (32 MB). Choose values that are significantly less than the address space limit. For Tru64 UNIX Version 5.0 and higher, be sure that you do not set the values too high unless it is recommended for your application.

### 3.2.6 Increasing the User Process Data Segment Size Limits

The `proc` subsystem attribute `per_proc_data_size` and `max_per_proc_data_size` specifies the maximum sizes of a user process data segment.

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 of the `per_proc_data_size` is 1,342,177,281 bytes. The default value of the `max_per_proc_data_size` is 1,073,741,824 (1 GB). The *recommended value* is 10,737,418,240 bytes (10 GB).

## 3.3 Modifying Virtual File System Attributes

You may be able to improve system performance by modifying the following virtual file system (`vfs`) kernel subsystem attributes for your version of the Tru64 UNIX operating system.

| Attribute               | Description   | 4.0F | 4.0G | 5.0 | 5.0A | 5.1 | 5.1A |
|-------------------------|---|------|------|-----|------|-----|------|
| <code>max_vnodes</code> | Specifies the size of the vnode cache (Section 3.3.1).  | Y    | Y    | Y   | Y    | Y   | Y    |
| <code>bufpages</code>   | Specifies the number of pages in the cache (Section 3.3.2).   | Y    | Y    | Y   | Y    | Y   | Y    |
| <code>bufcache</code>   | Specifies the percentage of physical memory that the kernel wires for the traditional buffer cache (Section 3.3.3). | Y    | Y    | Y   | Y    | Y   | Y    |

| Attribute | Description  | 4.0F | 4.0G | 5.0 | 5.0A | 5.1 | 5.1A |
|-----------|--|------|------|-----|------|-----|------|
| nchsize   | Specifies the size of the namei cache. (Section 3.3.4).            | Y    | Y    | Y   | Y    | N   | N    |
| nchsz     | Specifies the size of the namei cache hash table. (Section 3.3.5). | Y    | Y    | Y   | Y    | Y   | Y    |

**Note**

If you are running Tru64 UNIX Version 4.0F or 4.0G the kernel subsystem attributes are displayed using dashes ( - ); if you are running Tru64 UNIX Version 5.0 and higher the attributes are displayed using underscores ( \_ ).

For more information about the `vfs` subsystem attributes, see `sys_attrs_vfs(5)` for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

### 3.3.1 Increasing the Number of Open Files

The `vfs` subsystem attribute `max_vnodes` specifies the size of the vnode cache, which is always equal to or more than the maximum number of open files in the system.

You can increase the default value of this attribute to increase the maximum number of open files. You can also accomplish this task by increasing the value of the `proc` subsystem attribute `maxusers`.

The default value of the `max_vnodes` attribute is 5 percent of memory.

### 3.3.2 Specifying the Number of Pages in the Traditional Buffer Cache

The `vfs` subsystem attribute `bufpages` specifies the number of pages in the cache. You can reduce the memory allocated for the traditional buffer cache below 1 percent by decreasing this value.

The default value is calculated from the `bufcache` attribute.

### 3.3.3 Reducing the Size of the Traditional Buffer Cache

The `vfs` subsystem attribute `bufcache` specifies the percentage of physical memory that the kernel wires for the traditional buffer cache. If you have a large-memory system or if you use only AdvFS, you may want to decrease the size of the traditional buffer cache. The traditional buffer cache contains

recently accessed UFS, MFS, CD-ROM, or DVD-ROM file system metadata and block device (unmounted) I/O data. If you intend to use block devices with commands such as `dd`, `vdump`, or `vrestore`, you may want to increase the size of the buffer cache.

The default size of the traditional buffer cache is 3 percent of physical memory. For large-memory systems or systems that use only AdvFS, set the value of the `bufcache` attribute to 1 percent.

### 3.3.4 Increasing the Size of the Namei Cache

The `vfs` subsystem attribute `nchsize` specifies the number of `namei` cache entries in the cache, which caches pathname translation information (filename and `vnode` pair). If you have a large-memory system or increased `max_vnodes`, you may want to increase the `nchsize` attribute accordingly. The default value of the `namei` cache hash table is calculated from `nchsize`.

*For Tru64 UNIX Version 5.1 and higher:* You cannot modify the `nchsize` attribute; it is fixed at 15. The default value is calculated from the `nvnode` attribute. The `nvnode` attribute is equal to ten times the `maxusers` plus 148.

*For Tru64 UNIX Version 5.1 and lower:* The *recommended* value is half of the value of `max_vnodes` or larger.

### 3.3.5 Increasing the Size of the Namei Cache Hash Table

The `vfs` subsystem attribute `nchsz` specifies the size of the `namei` cache hash table, which is used to speed up the lookup of a specific `namei` cache entry. If you have a large-memory system or increased `nchsize` or `max_vnodes`, you may want to increase `nchsz` attribute accordingly.

The default size of the `namei` cache hash table is calculated from `nchsize`. The *recommended* value is the `max_vnode` attribute divided by 11.

*For Tru64 UNIX Versions 5.1 and higher:* You can only adjust the size of the `namei` cache by adjusting `nchsz`; each `namei` cache hash table can cache up to 15 `namei` cache entries. The default size of the `namei` cache hash table is calculated from the `nvnode` attribute.

## 3.4 Modifying Interprocess Communication Attributes

At boot time, the system allocates resources for System V shared memory and or System V semaphores. In some cases, database applications may require you to increase the System V shared memory and System V semaphores resource allocation.

This section describes how you may be able to improve performance by modifying the following Interprocess Communication (ipc) kernel subsystem attributes for your version of the Tru64 UNIX operating system:

| Attribute      | Description   | 4.0F | 4.0G | 5.0 | 5.0A | 5.1 | 5.1A |
|----------------|---|------|------|-----|------|-----|------|
| shm_max        | Specifies the maximum size of a single System V shared memory region (Section 3.4.1).   | Y    | Y    | Y   | Y    | Y   | Y    |
| shm_seg        | Specifies the maximum number of System V shared memory regions that can be attached to a single process at any point in time (Section 3.4.2). | Y    | Y    | Y   | Y    | Y   | Y    |
| shm_mni        | Specifies the maximum number of shared memory regions that can be used on the system at one time (Section 3.4.3).                             | Y    | Y    | Y   | Y    | Y   | Y    |
| ssm_thresh-old | Specifies the minimum size, in bytes, of a System V shared region for the use of shared page tables (Section 3.4.4).                          | Y    | Y    | Y   | Y    | Y   | Y    |
| sem_mni        | Specifies the maximum number of System V semaphores that can be used on the system at one time (Section 3.4.5).                               | Y    | Y    | Y   | Y    | Y   | Y    |
| sem_msl        | Specifies the maximum number of System V semaphores that can be used by a single process at one time (Section 3.4.6).                         | Y    | Y    | Y   | Y    | Y   | Y    |
| sem_opm        | Specifies the maximum number of operations that can be outstanding on a single System V semaphore at one time (Section 3.4.7).                | Y    | Y    | Y   | Y    | Y   | Y    |
| sem_ume        | Specifies the number of undo operations that can be outstanding on a single System V semaphore at one time (Section 3.4.8).                   | Y    | Y    | Y   | Y    | Y   | Y    |

---

**Note**

---

If you are running Tru64 UNIX Version 4.0F or 4.0G the kernel subsystem attributes are displayed using dashes ( - ); if you are running Tru64 UNIX Version 5.0 and higher the attributes are displayed using underscores ( \_ ).

---

For more information about `ipc` subsystem attributes see `sys_attrs_ipc(5)` for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

### 3.4.1 Increasing the System V Shared Memory Region

The `ipc` subsystem attribute `shm_max` specifies the maximum size of a single System V shared memory region. In some cases, memory-intensive database applications may benefit from a large system V shared memory region.

The default value of the `shm_max` attribute is 4,194,304 bytes (512 pages), which may be too small for a database application. If you increase the value, be sure that you have enough memory. For more information, see Section 3.2.2.

### 3.4.2 Increasing the Shared Memory Regions Attached to a Process

The `ipc` subsystem attribute `shm_seg` specifies the maximum number of System V shared memory regions that can be attached to a single process at any point in time.

The default value is 32. You may want to increase the value of the `shm_seg` attribute if the process' attempt to attach a shared memory region exceeds the limit. Increasing the value of the `shm_seg` attribute will consume memory if the process attaches many shared memory regions.

### 3.4.3 Increasing the Shared Memory Regions That Can be Used on the System at One Time

The `ipc` subsystem attribute `shm_mni` specifies the maximum number of shared memory regions that can be used on the system at one time. You may want to increase the maximum number of shared memory regions that can be used on the system at one time.

The default value of the `shm_mni` attribute is 100 (regions). The system rounds the number to the value associated with the next higher power of 2; for example, 128.

### 3.4.4 Increasing the Shared Page Table Limit

Third-level page table sharing occurs when the size of a System V shared memory segment, as created by the `shmget ( )` function, is equal to or larger than the value of the `ipc` subsystem attribute `ssm_threshold`.

You may want to increase the shared page table limit to restrict shared page tables to applications that create shared memory segments larger than 8 MB. However, if your application cannot use shared page tables because of alignment restrictions, you may want to disable the sharing of page tables.

The default value of the `ssm_threshold` attribute is 8,388,608 bytes (8 MB). Setting the `ssm_threshold` attribute to 0 (zero) disables the use of segmented shared memory.

### 3.4.5 Increasing System V Semaphores That Can be Used on the System at One Time

The `ipc` subsystem attribute `sem_mni` specifies the maximum number of System V semaphores that can be used on the system at one time. If your application can benefit from increasing the maximum number of System V semaphores, you may want to increase this value.

The default value is 10 (semaphores). The system rounds the number to the value associated with the next higher power of two.

### 3.4.6 Increasing System V Semaphores That Can be Used by a Single Process at One Time

The `ipc` subsystem attribute `sem_msl` specifies the maximum number of System V semaphores that can be used by a single process at one time. If your application can benefit from increasing the maximum number of System V semaphores that can be used by a single process at one time, you may want to increase this value.

The default value is 25 (semaphores).

### 3.4.7 Increasing Operations That Can Be Outstanding on a Single System V Semaphore at One Time

The `ipc` subsystem attribute `sem_opm` specifies the maximum number of operations that can be outstanding on a single System V semaphore at one time. If your application can benefit from increasing the maximum number of operations that can be outstanding on your system at one time you may want to increase this value.

The default value is 10 (operations).

### 3.4.8 Increasing the Number of Undo Operations That Can be Outstanding on a Single System V Semaphore at One Time

The `ipc` subsystem attribute `sem_ume` specifies the number of undo operations that can be outstanding on a single System V semaphore at one time. If you have a database server or a URL caching server, you may want to increase this value.

The default value is 10 (operations).

## 3.5 Modifying Virtual Memory Attributes

You may be able to improve large-memory system performance by tuning the following virtual memory (`vm`) kernel subsystem attributes for your version of the Tru64 UNIX operating system:

| Attribute                           | Description   | 4.0F | 4.0G | 5.0 | 5.0A | 5.1 | 5.1A |
|-------------------------------------|---|------|------|-----|------|-----|------|
| <code>ubc_maxpercent</code>         | Specifies the maximum amount of memory that can be allocated to the UBC (Section 3.5.1).  | Y    | Y    | Y   | Y    | Y   | Y    |
| <code>ubc_minpercent</code>         | Specifies the minimum percentage of memory that only the UBC can utilize (Section 3.5.1).   | Y    | Y    | Y   | Y    | Y   | Y    |
| <code>ubc_borrowpercent</code>      | Specifies the UBC borrowing threshold (Section 3.5.1).  | Y    | Y    | Y   | Y    | Y   | Y    |
| <code>vm_ubcseqpercent</code>       | Specifies the maximum amount of memory that can be used to cache a sequentially-accessed file (Section 3.5.2).                                      | Y    | Y    | Y   | Y    | Y   | Y    |
| <code>vm_ubcseqstart-percent</code> | Specifies the threshold value that determines when the UBC starts to check the percentage of UBC pages cached for each file object (Section 3.5.2). | Y    | Y    | Y   | Y    | Y   | Y    |
| <code>vm_page_free_target</code>    | Specifies the threshold value that stops paging (Section 3.5.3).  | Y    | Y    | Y   | Y    | Y   | Y    |
| <code>vm_page_free_reserved</code>  | Specifies the threshold value that determines when memory is limited to privileged tasks (Section 3.5.3).   | Y    | Y    | Y   | Y    | Y   | Y    |
| <code>vm_page_free_min</code>       | Specifies the threshold value that starts page swapping (Section 3.5.3).  | Y    | Y    | Y   | Y    | Y   | Y    |

| Attribute               | Description  | 4.0F | 4.0G | 5.0 | 5.0A | 5.1 | 5.1A |
|-------------------------|--|------|------|-----|------|-----|------|
| vm_page_free_optimal    | Specifies the threshold value that begins hard swapping (Section 3.5.4).   | Y    | Y    | Y   | Y    | Y   | Y    |
| vm_page_prewrite_target | Specifies the maximum number of pages that the vm subsystem will prewrite to swap space if it anticipates running out of memory (Section 3.5.5).   | Y    | Y    | Y   | Y    | Y   | Y    |
| vm_ubcdirtypercent      | Specifies the percentage of pages that must be dirty (modified) before the UBC starts writing them (Section 3.5.5).  | Y    | Y    | Y   | Y    | Y   | Y    |
| ubc_maxdirtywrite       | Number of I/O operations (per second) that the virtual memory subsystem performs when the number of dirty (modified) pages in the UBC exceeds the value of the vm-ubcdirtypercent attribute (Section 3.5.5). | Y    | Y    | Y   | Y    | Y   | Y    |

**Note**

If you are running Tru64 UNIX Version 4.0F or 4.0G the kernel subsystem attributes are displayed using dashes ( - ); if you are running Tru64 UNIX Version 5.0 and higher the attributes are displayed using underscores ( \_ ).

For more information about thevm subsystem attributes see `sys_attrs_vm(5)` for your version of the operating system:

[http://www.tru64unix.compaq.com/docs/pub\\_page/doc\\_list.html](http://www.tru64unix.compaq.com/docs/pub_page/doc_list.html)

### 3.5.1 Reducing the Amount of Memory Allocated to the UBC

The vm subsystem attribute `ubc_maxpercent` specifies the maximum amount of memory that can be allocated to the Unified Buffer Cache (UBC). Too much memory allocated to the UBC can cause excessive paging and swapping, which may degrade overall system performance. However, an insufficient amount of memory allocated to the UBC can degrade file system performance.

For large-memory systems, you may be able to free memory for use by processes by decreasing the memory allocated to the UBC. Be sure that decreasing the size of the UBC does not affect file-system performance or cause lock time-out panics. If you want to decrease the memory allocated to the UBC, start with the default value of the `ubc_maxpercent`

attribute (100) and decrease the value by decrements of 10. For example, on large-memory systems, you may want to use a value of 50 for the `ubc_maxpercent` attribute. Do not make the values of the `ubc_maxpercent` and `ubc_minpercent` attributes close together.

The `ubc_minpercent` attribute specifies the minimum percentage of memory that only the UBC can utilize. The remaining memory is shared with processes. The `ubc_maxpercent` attribute specifies the maximum percentage of memory that the UBC can utilize. The `ubc_borrowpercent` attribute specifies the UBC borrowing threshold.

Between the value of the `ubc_borrowpercent` attribute and the value of the `ubc_maxpercent` attribute, the memory that is allocated to the UBC is considered borrowed from processes. When paging begins, these borrowed pages are reclaimed first, until the amount of memory allocated to the UBC decreases to the value of the `ubc_borrowpercent` attribute.

The default value for `ubc_minpercent` is 10 percent. The default value for `ubc_maxpercent` is 100 percent. The default value for `ubc_borrowpercent` is 20 percent. If your disks are busy with file-system I/O and the system has sufficient free pages, use the default values.

If you have a low free page count, you may want to increase the memory available to processes by reducing the memory available to the UBC. Try to keep in memory the working set of your processes, even if it increases the number of UBC misses.

Reducing the borrowed memory threshold by decreasing the value of the `ubc_borrowpercent` attribute may improve the system response time when memory is low. However, this may also reduce UBC performance.

### 3.5.2 Specifying Sequential File Caching

The `vm` subsystem attribute `vm_ubcseqpercent` specifies the maximum amount of memory that can be used to cache a sequentially accessed file. The `vm_ubcseqpercent` limit prevents a single, sequentially accessed file from filling the UBC with pages that will not be immediately accessed. Instead, the system caches only a small portion of the file and uses a buffering technique that enables old pages to be reclaimed for new allocations.

If you reuse large sequentially accessed files, increasing the value of the `vm_ubcseqpercent` attribute may improve performance for these files.

The default value of the `vm_ubcseqpercent` attribute is 10 percent of memory allocated to the UBC. This attribute does not affect files that are randomly accessed.

*For Tru64 UNIX Version 5.0 and higher:* The `vm` subsystem attribute `vm_ubcseqstartpercent` specifies the size of the UBC as a percentage of

UBC memory, at which time the virtual memory subsystem starts stealing the Unified Buffer Cache least recently used (UBC LRU) pages for a file to satisfy the demand for pages. This value must be less than the value of the `ubc_maxpercent` attribute.

The default value of the `vm_abcseqstartpercent` attribute is calculated by dividing the value of the `ubc_maxpercent` attribute by two.

To force the system to reuse the pages in the UBC instead of taking pages from the free list, perform the following tasks:

1. Ensure that the maximum size of the UBC (`ubc_maxpercent`) is greater than the value of the `vm_abcseqstartpercent` attribute.
2. Ensure that the value of the `vm_abcseqpercent` attribute is greater than the referenced file.

In addition, if you have a large-memory system that is doing a lot of file-system operations, you may want to decrease the value of the `vm_abcseqstartpercent` attribute to 30 percent. Do not specify a lower value unless you also decrease the size of the UBC. If you decrease `vm_abcseqstartpercent`, do not change the value of the `vm_abcseqpercent` attribute.

### 3.5.3 Increasing the Paging Threshold

Paging begins when the number of pages on the free list is less than the value of the `vm` subsystem attribute `vm_page_free_target`. If you have a large amount of memory, or if your system experiences performance problems when memory is low, you may want to increase the paging threshold. Do not modify this value if you are not paging.

The following table provides recommendations for the `vm_page_free_target` attribute for different memory configurations:

| Size of Memory     | Value of <code>vm_page_free_target</code> |
|--------------------|---|
| Up to 512 MB       | 128                                       |
| 513 MB to 1024 MB  | 256                                       |
| 1025 MB to 2048 MB | 512                                       |
| 2049 MB to 4096 MB | 768                                       |
| More than 4096 MB  | 1024                                      |

If the number of pages falls below the value of the `vm_page_free_reserved` attribute, the system may hang. The default value is 10 pages. If your system experiences performance problems when memory is low, you may want to increase the value of the `vm_page_free_reserved` attribute to 64.

Do not make the value of the `vm_page_free_reserved` higher than the value of the `vm_page_free_min` attribute.

### 3.5.4 Tuning Swapping Threshold

Hard swapping begins when the number of pages on the free list is less than the value of the `vm` subsystem attribute `vm_page_free_optimal` for five seconds. Because swapping has a drastic impact on system performance, large-memory systems and systems running large programs should avoid swapping. The best way to avoid swapping is to minimize paging. However, you can tune the swapping threshold to optimize either response time or throughput. You do not need to modify the swapping threshold if the system is not swapping processes.

To increase throughput, increase the value of the `vm_page_free_optimal` attribute. As more processes are swapped out, fewer processes are actually executing and more work is done. However, increasing the rate of swapping may degrade interactive response time. To improve interactive response time, decrease the value of the `vm_page_free_optimal` attribute. Decreasing the rate of swapping will degrade throughput.

Increase or decrease the value of the `vm_page_free_optimal` attribute by only two pages at a time. Do not specify a value that is more than the value of the `vm` subsystem attribute `vm_page_free_target`.

### 3.5.5 Increasing the Rate of Modified Page Prewriting

The `vm` subsystem attempts to prevent a memory shortage by prewriting modified (dirty) pages to disk. To reclaim a page that has been prewritten, the `vm` subsystem only needs to validate the page, which can improve performance.

You do not need to tune modified page prewriting if the system is not paging. You can increase the rate of modified page prewriting to prevent a drastic performance degradation when memory is exhausted, or you can decrease the rate of modified page prewriting to improve peak workload performance.

When the virtual memory subsystem anticipates that the pages on the free list will soon be depleted, it prewrites to disk the oldest modified inactive and UBC LRU pages. This number is specified by the `vm` subsystem attribute `vm_page_prewrite_target`.

The following table describes tuning recommendations for modifying page prewriting:

| <b>Goals</b>  | <b>Tuning Recommendations</b>  |
|---|--|
| <b>To increase the rate of inactive modified page prewriting.</b>   | Increase the default value of the <code>vm_page_prewrite_target</code> attribute by increments of 64 pages.  |
| <b>To increase the rate of UBC LRU modified page prewriting.</b>  | Decrease the default value of the <code>vm_ubcdirtypercent</code> attribute by decrements of 1 percent.  |
| <b>To improve peak workload performance.</b>  | Decrease the rate of modified page prewriting. However, this will cause a drastic performance degradation when memory is exhausted.  |
| <b>To decrease the rate of inactive modified page prewriting.</b>   | Decrease the default value of the <code>vm_page_prewrite_target</code> attribute.  |
| <b>To decrease the rate of UBC LRU modified page prewriting.</b>  | Increase the default value of the <code>vm_ubcdirtypercent</code> attribute.   |
| <b>To minimize the impact of I/O spikes caused by the sync function.</b>  | Increase the value of the <code>vm</code> subsystem attribute <code>ubc_maxdirtywrites</code> .  |
| <b>To prevent a drastic performance degradation when memory is exhausted, and provide better file system integrity.</b> | You can increase the rate of modified page prewriting. However, this will increase the amount of continuous disk and reduce peak workload performance. Use the <code>iostat</code> command to display disk I/O statistics (see Section 1.3). |