PARSEC Group

Our Trainers Consult. Our Consultants Train.

hp Tru64 Performance

Presented by

Jamie Bridge

www.parsec.com | 888-4-PARSEC

Introduction

Operating system performance is a diverse, sometimes multi-faceted issue; in other words, issues related to OS performance may not lie in one particular area. **Tru64 UNIX** is a good example of an OS that displays this attribute; not only does it have an **extensive suite of kernel subsystem attributes** that can be tuned and altered for a variety of purposes, but the various Alpha hardware architectures that Tru64 UNIX runs on can sometimes help or perhaps even hinder performance as well.

This webinar will discuss Tru64 UNIX performance and some recommended steps to take in order to identify, isolate and remedy certain kinds of performance problems. We will also look at some of the performance analysis tools available for the Tru64 UNIX operating system, and how to get the most out of them for whatever particular performance issue is being dealt with.

Common Concerns

Some common areas of concern for Tru64 UNIX (not all of which will be discussed in the next hour):

- **3rd Party Application performance/responsiveness** (Oracle, Sybase, Apache, etc.)
- **Operating System performance/responsiveness** (NFS, cron, AdvFS, process utilization, etc.)
- **Layered Product performance/responsiveness** (TruCluster, Webes, Java VM, Insight Manager, etc.)
- **Hardware component performance/responsiveness** (disk i/o, tape i/o, host bus adapter throughput, etc.)

Performance Issues

The performance areas to be discussed in the next hour are as follows:

- Identifying & isolating OS performance issues
- Kernel subsystem performance/responsiveness
- Isolating file-system component issues (NFS, AdvFS)

Common Performance Scenarios

Common performance scenarios:

- System performance is “slow” overall
- 3rd Party or Layered Product App’s performing slowly
- Certain Tru64 UNIX commands not executing quickly
- Overall OS functionality is hindered (daemons not running, poor application response, etc.)
- File-system performance issues (data not accessible, slow i/o rates, hardware-induced issues, etc.)

Addressing System Performance

Questions to ask regarding system performance:

- What exactly about the system is “slow”?
- Is this a system-wide problem or an isolated problem?
- What is the resource load on the system?
- When did the performance degradation start?
- Is it a consistent problem, or does it only occur at a certain time?
- What events (if any) led up to the performance problem?
- What has been done to fix the performance problem?

Virtual Memory

Virtual Memory: A means of extending the capabilities of physical memory. Tru64 UNIX uses a combination of physical memory and disk space to create virtual memory; this allows for the creation of much larger process regions, as virtual memory can support more processes than physical memory alone. **Knowing how virtual memory is being utilized can help isolate a system performance issue.**

Virtual Memory Page: The basic unit of physical and virtual memory **(1 page=8192 bytes).** Virtual memory tries to keep recently referenced virtual pages for a process in physical memory. When a process makes reference to virtual pages, they are brought into physical memory from their storage locations on disk. **Keeping processes as memory resident as possible helps achieve a higher level of system performance.**

Use the **“vmstat”** command to get an idea of the virtual memory resource load on the system in question.

vmstat

Example output from “vmstat”:

```
root@marquis in / --> vmstat 1 5
(1=seconds; 5=lines of output)

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
 3 212  38  17K 6590 5361  427K  87K 149K   735  98K   37  10  54  95  0  0 100
 3 212  38  17K 6588 5361     3   15   23     0   29    0   5  77  84  0  1  99
 3 212  38  17K 6588 5361     0    0    0     0    0    0   3  97  86  0  0 100
 3 212  38  17K 6588 5361     0    0    0     0    0    0   3  40  80  0  0 100
 3 212  38  17K 6588 5361     0    0    0     0    0    0   9  40 105  0  0 100
```

vmstat Fields

Fields from “vmstat” output:

Process information:

r Number of threads that are running or are runnable.

w Number of threads waiting interruptibly.

u Number of threads waiting uninterruptibly.

vmstat Fields

Fields from “vmstat” output (cont.):

Virtual memory information:

act

Total number of pages on the active list, the inactive list (pages that are allocated but are most likely to be used for paging), and UBC least recently used (LRU) list.

free

Total number of pages that are clean and available for use.

wire

Total number of pages that are currently in use and cannot be used for paging (not a real list).

fault

Number of address translation faults that have occurred.

cow

Number of copy-on-write page faults, which occur if the requested page is shared by a parent process and one or more child processes (using the fork function) and if one of the processes needs to modify the page. In this case, VM loads a new address into the translation buffer and copies the contents of the requested page into the new address for modification by the process.

vmstat Fields

Fields from “vmstat” output (cont.):

Virtual memory information (cont.):

zero

Number of zero-filled-on-demand page faults, which occur if VM cannot find the page in the internal data structures and if the requested page is new and has never been referenced. In this case, VM initializes a physical page (the contents of the page are zeroed out) and loads the address into the page table.

react

Number of pages that have been faulted while on the inactive list.

pin

Number of requests for pages from a pager.

pout

Number of pages that have been paged out.

vmstat Fields

Fields from “vmstat” output (cont.):

Interrupt information:

in Number of nonclock device interrupts per second.
sy Number of system calls called per second.
cs Number of task and thread context switches per second.

CPU information:

us Percentage of user time for normal and priority processes.
sy Percentage of system time.
id Percentage of idle time.

iowait

Percentage of iowait. If the **-w** option is not specified, the iowait time is included in the id statistic.

vmstat Examples

- Specify "vmstat -f" to display fork statistics only.
- Specify "vmstat -s" for a single display of accumulated statistics, as well as page size.

```
Examples:
========
#vmstat -f
Fork statistics:
     12639    forks
      1595    vforks

#vmstat -s
Virtual Memory Statistics: (pagesize = 8192)
      4010   active pages
      7019   inactive pages
     39617   free pages
     10505   wired pages
   1888495   virtual memory page faults
    313876   copy-on-write page faults
    953390   zero fill page faults
       388   reattaches from reclaim list
    318189   pages paged in
         0   pages paged out
 150950465   task and thread context switches
   4618056   device interrupts
1060849232   system calls
```

vmstat Procedures

Use the “–P” switch to vmstat to get virtual and physical memory utilization information:

```
root@marquis in / --> vmstat -P

Total Physical Memory =   256.00 M
                      =    32768 pages
Physical Memory Clusters:

start_pfn     end_pfn       type  size_pages / size_bytes
      0        256        pal         256 /    2.00M
    256        32663        os      32407 /  253.18M
  32663        32768        pal       105 /  840.00k
…………
```

Understanding Swap Space

Swap space: The region on disk that houses modified virtual memory **(8K)** pages. Modified virtual pages can be moved to swap space if the physical pages (pages in physical memory) that contain the virtual pages are needed by either a newly referenced virtual page or by a page with a higher priority. **Tru64 UNIX** uses 2 means of moving virtual pages between physical memory and disk:

Paging: Moves individual virtual pages between disk and physical memory.

Swapping: Moves large number of virtual pages (i.e. pages that are associated with a particular process or group of processes) between physical memory and disk.

Paging Types

Types of paging:

page-in: If a process references a virtual page that is not in physical memory, Tru64 UNIX reads a copy of the virtual page from its location on **disk** or **swap space** into physical memory.

page-out: If a physical page is needed to hold a newly-referenced virtual page or a page with a higher priority, Tru64 UNIX writes a modified virtual page/group of pages that has not been recently referenced to **swap space**.

page-fault: If a requested address for a page is not active in the page table, the virtual memory (**vm**) subsystem locates the page and performs a virtual-to-physical address translation in the page table.

The rate of page-ins and page-outs on a system may provide evidence of a performance issue (if one exists) and can be monitored with the “**vmstat**” command.

Swap Modes

Swap modes:

Immediate (eager) mode: Each modifiable virtual page is assigned a page of swap space when created.

Deferred (lazy) mode: Swap space is not allocated until the system needs to write a modified virtual page to swap space.

The Tru64 UNIX parameter that governs which swap algorithm is used is “**vm: vm_swap_eager**”:

```
 --> sysconfig -q vm vm_swap_eager
vm:
vm_swap_eager = 1←(1=on, 0=off)
```

Swap Space Example

Some reasons for adding additional swap space:

- Adding or upgrading a memory-intensive application(s)
- System is becoming memory-restricted (per vmstat)
- Tru64 UNIX presents warnings to syslogd about swap space
- Comply with the “**swap=2 to 3x physmem”** rule for moderate memory systems

Real world example of when one might want to add swap:

You may see the following message in /var/adm/messages:

```
Jun 17 07:41:08 marquis vmunix: swap space below 10 percent free
Jun 19 05:55:59 marquis vmunix: swap space below 10 percent free
Jun 21 06:58:51 marquis vmunix: swap space below 10 percent free
```

………..

Swap Space Example (cont.)

Resolution:

1. **Find out what disks you can use for additional swap:**

```
#readlabelsv5 (script to quickly identify what disks are not in use)

Total disks to be read: 7

dsk0: 4 partitions not in use.
dsk1: 8 partitions not in use. <--Not in use by a filesystem
dsk4: 7 partitions not in use.
dsk5: 7 partitions not in use.
dsk6: 8 partitions not in use. <--Not in use by a filesystem
dsk3: 8 partitions not in use. <--Not in use by a filesystem
dsk2: 7 partitions not in use.
```

Swap Space Example (cont.)

2. Check the disklabel:

```
#disklabel -r dsk1

..........

8 partitions:
#          size      offset    fstype  fsize  bsize   cpg  # ~Cyl values
  a:     131072           0    unused      0      0        #     0 - 57*
  b:     262144      131072    unused      0      0        #   57*- 173*
  c:    8380080           0    unused                      #     0 - 3707
  d:          0           0    unused      0      0        #     0 - 0
  e:          0           0    unused      0      0        #     0 - 0
  f:          0           0    unused      0      0        #     0 - 0
  g:    3993432      393216    unused      0      0        #  173*- 1940*
  h:    3993432     4386648    unused      0      0        # 1940*- 3707
```

Swap Space Example (cont.)

3. Mark the "c" partition (whole disk) as used for swap:

```
#disklabel -sF dsk1c swap
```

4. Add this to the /etc/sysconfigtab file:

```
..........

swapdevice = /dev/disk/dsk0b,/dev/disk/dsk1c
```

5. Turn on the paging files in multi-user mode:

```
#swapon -a
```

Swap Space Example (cont.)

6. Confirm that the new swap partition is being used:

```
#swapon -s

Swap partition /dev/disk/dsk0b (default swap):
   Allocated space:        49152 pages (384MB)
   In-use space:             844 pages (  1%)
   Free space:             48308 pages ( 98%)

Swap partition /dev/disk/dsk1c:                <--
   Allocated space:       523755 pages (4.00GB)
   In-use space:               1 pages (  0%)
   Free space:            523754 pages ( 99%)


Total swap allocation:
   Allocated space:       572907 pages (4.37GB)  <--
   Reserved space:         15566 pages (  2%)
   In-use space:             845 pages (  0%)
   Available space:       557341 pages ( 97%)
```

PARSEC Group
Our Trainers Consult. Our Consultants Train.

Swap Space Example (cont.)

In order for these settings to survive a reboot, do the following:

1.

```
#sysconfig -Q vm swapdevice
vm:
swapdevice -    type=STRING op=CQ min_len=0 max_len=1024
```

op=CQ: If no "R" is present, then the parameter is not configurable at run-time.

2.

```
#sysconfig -q vm >stanza-file
```

3.

```
#vi stanza-file
.........

swapdevice = /dev/disk/dsk0b,/dev/disk/dsk1c
```

(save work in vi)

4.

Merge the changes from the stanza-file into the **vm** kernel subsystem (discussed later):

```
#sysconfigdb -m -f stanza-file vm
```

As a result, the changes made will persist across reboots.

PARSEC Group
Our Trainers Consult. Our Consultants Train.

Answering Performance Questions

How do I determine where the slowdown is occuring?

What tend to be the most common areas here are with regards to **commands** and **applications**. Commands that normally return quickly no longer do so; applications that complete execution in a certain time interval now take longer to complete.

If the System Administrator documents how the system performs under normal circumstances and has some understanding of how commands and applications interact with Tru64 UNIX, then the OS tools available will go that much further in isolating the problem.

Answering Performance Questions

Some commands and files to use to help isolate slow system performance:

1. **vmstat** (discussed earlier; details virtual memory performance statistics)
2. **iostat** (details disk i/o performance statistics)
3. **netstat** (details network performance statistics)
4. **ps** (details information about system processes)
5. **sysconfig** (means of viewing and modifying kernel subsystem parameters)
6. **nfsstat** (details Network File System performance statistics)
7. **advfsstat** (details Advanced File System performance statistics)
8. **volstat** (details Logical Storage Manager (LSM) performance statistics)
9. **syslogd** (system event logging daemon; ref. the /var/adm/syslog.dated directory)
10. **evmd** (system event logging daemon; ref. the /var/evm/evmlog directory)
11. **cpustat** (cpu performance statistics; function of the "kdbx" kernel debugger)
12. **df** (details mounted filesystem statistics; starting point for further FS analysis)
13. **du** (details disk usage statistics)
14. **dumpsys** (means of generating a system/memory dump on a running system)
15. **sys_check** (data-gathering tool to help provide more detailed configuration information)
16. **collect** (statistical reporting and gathering tool)
17. **spike** (performs code optimization after linking a program)
18. **Force a system crash** in order to capture the state of the system while it is running slow

Answering Performance Questions

How can I isolate the problem?

Questions to help narrow down the scope of the issue:

- How long has the problem been around?
- How many users are seeing the problem?
- Are all aspects of the system impacted by the problem?
- Does it occur at a specific time, or all of the time?
- What (if any) changes were made on the system?
- What analysis has been done to date?

Answering Performance Questions

How do I determine the resource load on the system?

Use “**iostat**” to get disk performance statistics:

```
root@marquis in / --> iostat dsk4 1 5 (5 lines of output in 1-second intervals)
    tty      floppy0          dsk4          cpu
 tin tout    bps    tps    bps    tps  us ni sy id
   0    2      0      0      9      0   0  0  0100
   0   52      0      0      0      0   0  0  0100
   0   52      0      0      0      0   0  0  0100
   0   52      0      0      0      0   0  0  0100
   0   52      0      0      0      0   0  0  0100

root@marquis in / --> iostat dsk5 1 5
    tty      floppy0          dsk5          cpu
 tin tout    bps    tps    bps    tps  us ni sy id
   0    2      0      0      1      0   0  0  0100
   0   52      0      0      0      0   0  0  0100
   0   52      0      0      0      0   0  0  0100
   0   52      0      0      0      0   0  0  0100
   0   52      0      0      0      0   0  0  0100
```

PARSEC Group
Our Trainers Consult. Our Consultants Train.

Answering Performance Questions

Use **“netstat”** to get network performance statistics:

```
root@marquis in / --> netstat
Active Internet connections
Proto Recv-Q Send-Q  Local Address              Foreign Address            (state)
tcp        0      0  marquis.parsec.com.7920   marquis.parsec.com.1032   ESTABLISHED
tcp        0      0  marquis.parsec.com.1032   marquis.parsec.com.7920   ESTABLISHED
tcp        0      2  marquis.parsec.com.telnet parsec-vshh7yzx.parsec.com.1765 ESTABLISHED

………………
```

When used without options, the **netstat** command displays a list of active sockets for each protocol. The default display shows the following items:

+ Local and remote addresses
+ Send and receive queue sizes (in bytes)
+ Protocol
+ State

Address formats are of the form host.port or network.port if a socket's address specifies a network but no specific host address. The host and network address are displayed symbolically unless -n is specified.

PARSEC Group
Our Trainers Consult. Our Consultants Train.

Answering Performance Questions

Use **“netstat –I ifname”** to get statistics on a specific network interface:

root@marquis in / --> netstat -I nr0 1 **(1=1 second intervals)**

```
   input    (nr0)  output           input   (Total)    output
 packets  errs  packets  errs colls   packets  errs  packets   errs colls
    5047     0      682       0     0     15277     0     2168     0    44
       4     0        1     0     0        10     0        2     0     0
       3     0        1     0     0         8     0        2     0     0
       1     0        1     0     0         2     0        2     0     0
       1     0        1     0     0         2     0        2     0     0
       2     0        1     0     0         5     0        2     0     0
       4     0        1     0     0        11     0        4     0     0
       3     0        1     0     0         8     0        2     0     0
       2     0        1     0     0         5     0        2     0     0
       1     0        1     0     0         2     0        2     0     0
```

PARSEC Group
Our Trainers Consult. Our Consultants Train.

Answering Performance Questions

Use “**netstat –I ifname –s**” to display additional (data-link layer) stats for a specified network interface:

```
root@marquis in / --> netstat -I nr0 -s

nr0 Ethernet counters at Thu Jun 23 11:09:25 2005

           5326 seconds since last zeroed
         604404 bytes received
         155112 bytes sent
           6482 data blocks received
           1023 data blocks sent
         515096 multicast bytes received
           5351 multicast blocks received
          11583 multicast bytes sent
            244 multicast blocks sent
              1 blocks sent, initially deferred
             20 blocks sent, single collision
             20 blocks sent, multiple collisions
              0 send failures
              0 receive failures
```

PARSEC Group
Our Trainers Consult. Our Consultants Train.

Answering Performance Questions

Use the **“ps”** command to gather detailed information about system processes:

```
root@marquis in / --> ps auxw | more

USER        PID %CPU %MEM   VSZ  RSS TTY     S    STARTED        TIME COMMAND
root       1518  0.0  0.0 2.64M  88K pts/0   R  + 12:08:45    0:00.00 –ksh (ksh)
root       1517  0.0  1.6 6.45M 4.0M pts/0   R  + 12:08:45    0:00.05 ps auxw
root       1496  0.0  0.1 2.64M 344K pts/0   S    12:08:34    0:00.03 –ksh (ksh)
jbridge    1492  0.0  0.2 2.82M 432K pts/1   S  + 12:08:01    0:00.04 more messages
jbridge    1477  0.0  0.1 2.61M 320K pts/1   I    12:06:12    0:00.04 –ksh (ksh)
root       1475  0.0  0.3 10.7M 872K ??      S    12:06:01    0:00.12 /usr/sbin/sshd2
```

PARSEC Group
Our Trainers Consult. Our Consultants Train.

Answering Performance Questions

Use the **“ps”** command to gather detailed information about system processes **(cont.):**

```
#ps -Amo pid,ppid,rssize,comm,state,cputime,pcpu,psr | more

PID      PPID  RSS COMMAND        S         TIME %CPU  PSR
0          0  18M kernel idle    R <     0:11.83  0.0   ~0
1         0  96K init            IL     0:00.10  0.0    0
3         1 1.0M kloadsrv        I      0:00.07  0.0    0
5         1 352K hotswapd        S      0:00.00  0.0    0
54          1 328K esmd          I      0:00.03  0.0    0
65          1 112K update        S       0:00.00  0.0    0
190           1 1.0M evmd          I     0:00.46  0.0    0
227         190 456K evmlogger  I       0:00.22  0.0    0
228         190 360K evmchmgr  I        0:00.03  0.0    0
355           1 176K niffd          I     0:00.28  0.0    0
```

Answering Performance Questions

Use the “**cpustat**” function of the **kdbx** debugger to generate cpu usage statistics (must be the root user to use this command):

```
root@marquis in / --> echo "cpustat -update 1“ | kdbx -k /vmunix
```

(report statistics every second)

```
Cpu   User (%)    Nice (%) System (%)  Idle (%)   Wait (%)
===== ========== ========== ========== ========== ==========
   0      0.24       0.13       0.33      99.08       0.23
   0     41.74       0.00      58.26       0.00       0.00
   0     40.36       0.00      59.64       0.00       0.00
   0     39.31       0.00      60.69       0.00       0.00
   0     38.32       0.00      61.68       0.00       0.00
   0     38.19       0.00      61.81       0.00       0.00
```

(<ctrl>C’d out after 6 reports)

```
pid 1931 exited with signal SIGINT


root@marquis in / -->
```

Kernel Tuning

While a number of Tru64 UNIX commands can report back on system resource load, the **Tru64 UNIX kernel subsystem infrastructure** can actually control how the system handles this load. The **sysconfig** command is one way to increase or decrease the value of kernel subsystem parameters, should the situation call for it.

There are **over 80 kernel subsystems in the Tru64 UNIX (v5.1B) Operating System**. Not all of these subsystems need or require administrator interaction, though; of the 80+ subsystems in Tru64 UNIX (V5.1B), **administrators typically only deal with between 5 to 10 of them** when performing day-to-day administration tasks.

When discussing Tru64 UNIX performance, at some point the topic of **kernel tuning** more than likely will come up. What follows will detail some of the more common areas of kernel tuning that can have a noticeable impact on Tru64 UNIX performance.

Kernel Tuning

Why tune kernel subsystems?

- Can help the system perform faster/more efficiently
- Can help applications perform faster/more efficiently
- Resource limits have been exceeded (via syslogd)
- Technical documentation says to tune
- The “sys_check” utility (discussed later) says to tune
- System configuration (HW/SW) has changed
- System workload has changed

Kernel Tuning

Some of the more common kernel subsystems that may need to be tuned on a routine basis:

```
root@marquis in / --> sysconfig –s | more

generic: loaded and configured
inet: loaded and configured
ipc: loaded and configured
proc: loaded and configured
socket: loaded and configured
vm: loaded and configured
```

Kernel Tuning Reference

Reference:

1.
If you ever need to know what tunable parameters exist for a given subsystem, or need/want a general description of what a parameter does, reference the man page for that subsystem as follows: **#man sys_attrs_subsysname**

Example:
#man sys_attrs_vm
sys_attrs_vm - system attributes for the vm kernel subsystem

2.
Reference the **Tru64 UNIX Tuning Guide** for your operating system version at the following location on the WWW:

http://h30097.www3.hp.com/docs/pub_page/doc_list.html

VM Subsystem

As mentioned before, **virtual memory** is a means of extending the capabilities of **physical memory**. Given the importance of this feature, the “**vm**” subsystem is (arguably) the most frequently altered kernel subsystem.

Here is how to view parameters in the **vm** subsystem (or any subsystem for that matter):

```
 --> sysconfig -q vm
vm:
ubc_minpercent = 10
ubc_maxpercent = 100
ubc_borrowpercent = 20
vm_max_wrpgio_kluster = 32768
vm_max_rdpgio_kluster = 16384

………………
```

Unified Buffer Cache (UBC)

Unified Buffer Cache (UBC): Means of retaining actual file data (reads and writes from conventional file activity) for faster data retrieval. **UBC** and the **virtual memory subsystem** both share and compete for all of main memory.

ubc_minpercent: Minimum % of physical memory that the UBC can use.
Default value: **10** (percent)
Minimum value: 0
Maximum value: **100** (percent)

ubc_maxpercent: Maximum % of physical memory that the UBC can use at one time.

Default value: 100 (percent)
Minimum value: 0
Maximum value: 100

NOTE RE: ubc_maxpercent (T64 v5.1B): It is recommended that this value be set to a value in
the range of 70 to 80 percent. On an overloaded system, values higher than 80 can delay return of excess UBC pages to vm and adversely affect performance.

(Source: "man sys_attrs_vm", T64 v5.1B)

PARSEC Group
Our Trainers Consult. Our Consultants Train.

“proc” Subsystem

Another subsystem to keep in mind with regards to enhancing system performance is the “**proc**” subsystem. This subsystem, as the name implies, allows for the efficient management of user and system processes in the Tru64 UNIX Operating System.

The command to view **proc** subsystem parameters is as follows:

```
--> sysconfig -q proc
proc:
max_proc_per_user = 256
max_threads_per_user = 1024
per_proc_stack_size = 8388608
max_per_proc_stack_size = 33554432
per_proc_data_size = 134217728
max_per_proc_data_size = 1073741824
max_per_proc_address_space = 4294967296
per_proc_address_space = 4294967296
```

“maxusers” Parameter

The “**maxusers**” parameter in the **proc** subsystem:

Controls the number of simultaneous users that a system can support without straining system resources. System algorithms use the **maxusers** value to size various system data structures and to determine the amount of space allocated to system tables, such as the system process table. In a situation where multiple users are complaining of slow system performance, you may want to consider raising the value of the **maxusers** parameter.

Default value: System dependent
Minimum value: 8 (users)
Maximum value: 16,384

Increasing the value of the **maxusers** attribute allocates more system resources to the kernel. However, it also increases the amount of physical memory consumed by the kernel. Changing the value of the **maxusers** attribute automatically adjusts the values of other attributes, including the **taskmax**, **threadmax**, and **min_free_vnodes** attributes if you have not explicitly overridden the default values for those attributes.

(Source: “man sys_attrs_proc”, T64 v5.1B)

PARSEC Group
Our Trainers Consult. Our Consultants Train.

“maxusers” Parameter

A template to follow when considering a change to the **maxusers** parameter:

Size of Memory	Value of maxusers
Up to 256 MB	128
257 MB to 512 MB	256
513 MB to 1024 MB	512
1025 MB to 2048 MB	1024
2049 MB to 4096 MB	2048
4097 MB or more	2048

(Source: Tru64 UNIX 5.1B System Configuration and Tuning Manual)
(WWW: http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ARH9GCTE/TITLE.HTM)

PARSEC Group
Our Trainers Consult. Our Consultants Train.

“proc” Subsystem

While the **proc** kernel subsystem governs overall system process management, the **limit (csh) and ulimit (ksh)** commands affect the current execution (shell) environment. Once a limit has been decreased using ulimit, only a user with root privileges may increase it.

If you are seeing errors or performance issues that look to be specific to a particular execution environment, use the **ulimit** command to check and set current execution environment settings:

```
root@marquis in / --> ulimit
unlimited

root@marquis in / --> ulimit –a (Lists all current resource limits)
time(seconds)        unlimited
file(blocks)         unlimited
data(kbytes)         131072
stack(kbytes)        8192
memory(kbytes)       238288
coredump(blocks)     0
nofiles(descriptors) 4096
vmemory(kbytes)      4194304
```

(See “man ulimit” for details on increasing or decreasing these parameters)

Isolating File System Component Issues

Some common file-system components within Tru64 UNIX that can be subject to system performance degradation:

NFS=Network File System; means of sharing files and filesystems across a network.
AdvFS=Advanced File System; default filesystem for /, /usr and /var in Tru64 UNIX V5.

NFS:

- Shares the Unified Buffer Cache (UBC) with the **vm** subsystem & local filesystems.
- Network-based; NFS performance can suffer in a heavily used network.

AdvFS:

- Very tunable in many ways (de-fragmentation, balancing of data, file-set quotas)
- Very expandable (100 active domains, 250 volumes per domain, unlimited file-sets)
- Performance can suffer if all of this flexibility is not administered efficiently

Isolating File System Component Issues

Some questions to ask regarding **NFS** performance issues:

- How congested is the network as a whole?
- Are sufficient resources being allocated to NFS?
- Is the problem on the NFS client or server?

Some questions to ask regarding **AdvFS** performance issues:

- How many domains and file-sets constitute the file-system?
- What are the I/O statistics reporting?
- How fragmented is the file-system in question?
- Is the file-system made up of one or more volumes?

NFS Performance Issues

Some ways to isolate NFS performance issues:

- Use the “**netstat**” command to identify network performance bottlenecks
- Use the “**nfsstat**” command to get NFS performance statistics
- Use the “**nfswatch**” command to monitor NFS server activity **(requires additional kernel configuration; i.e. “options PACKETFILTER” in /sys/conf/HOSTNAME)**
- Monitor **syslogd** files for errors/problems **(/var/adm/syslog.dated, /var/adm/messages**)

Examples of NFS-related errors in /var/adm/messages:

```
Jun 17 21:27:34 beagle vmunix: NFS3 server marquis.parsec.com not responding still trying
Jun 17 21:27:34 beagle vmunix: NFS3 server marquis.parsec.com ok
Jun 18 23:51:42 beagle vmunix: fcntl: Local lock-manager not registered
Jun 18 23:51:42 beagle vmunix: fcntl: Local lock-manager not registered
Jun 18 23:57:03 beagle vmunix: fcntl: Local lock-manager not registered
```

NOTE: NFS performance can suffer if file-locking is in use on an NFS file. The locks prevent the file from being cached on the client.

PARSEC Group
Our Trainers Consult. Our Consultants Train.

“nfsstat” Command

Example of the “nfsstat” command:

```
root@marquis in / --> nfsstat

Server rpc:
tcp:      calls   badcalls   nullrecv    badlen    xdrcall    creates
            0         0          0          0         0         64
udp:      calls   badcalls   nullrecv    badlen    xdrcall
          3770        0          0          0         0

Server nfs:
     calls   badcalls    badprog    badproc    badvers    badargs
     3770        0          0          0          0          0
 unprivport   weakauth
       0          0
```

(See “man nfsstat” for more details)

“nfswatch” Command

Example of the “nfswatch” command:

```
root@beagle in / --> nfswatch
NFSWATCH Version 4.1 of 1 December 1993
Watch packets from all hosts to beagle.parsec.com on Ethernet interface tu0;
log to "nfswatch.log" (logging off);
snapshots to "nfswatch.snap";
cycle time 10 seconds...
beagle.parsec.com          Sun Jun 26 23:23:02 2005   Elapsed time:   00:00:10
Interval packets:        10 (network)        10 (to host)         0 (dropped)
Total packets:           10 (network)        10 (to host)         0 (dropped)
                    Monitoring packets from interface tu0
                   int    pct   total                       int    pct    total
ND Read                0     0%        0 TCP Packets            0     0%        0
ND Write               0     0%        0 UDP Packets            2    20%        2
NFS Read               0     0%        0 ICMP Packets           0     0%        0
NFS Write              0     0%        0 Routing Control        0     0%        0
NFS Mount              0     0%        0 Address Resolution     1    10%        1
YP/NIS/NIS+            0     0%        0 Reverse Addr Resol     0     0%        0
RPC Authorization      0     0%        0 Ethernet/FDDI Bdcst    9    90%        9
Other RPC Packets      0     0%        0 Other Packets          7    70%        7
                             4 file systems
    File Sys        int   pct   total       File Sys        int   pct   total
/                    0     0%       0
/tmp                 0     0%       0
/usr                 0     0%       0
/var                 0     0%       0
```

PARSEC Group
Our Trainers Consult. Our Consultants Train.

NFS Performance

Options in “mount” that can affect NFS (client) performance:

```
root@beagle in /var/adm --> mount -l -t nfs
/var/cust_files@marquis.parsec.com on /var/custl_files type nfs (v3, rw, exec, suid, dev,
nosync, noquota, grpid, atimes, udp, hard, intr, ac, cto, wsize=49152, rsize=49152,
timeo=11, maxtimo=20, retrans=4, acregmin=3, acregmax=60, acdirmin=30,
acdirmax=60)
```

- **rsize**=n (Sets the read buffer size to n bytes.)
- **wsize**=n (Sets the write buffer size to n bytes.)
- **timeo**=n (Sets the initial NFS timeout period for UDP mounts to n tenths of a second.)
- **maxtimo**=n (Sets the maximum value, in seconds, that is allowed between request
- transmissions. **UDP mounts only.**)
- **retrans**=n (Sets the number of NFS retransmissions to n.)
- **acregmin**=n (Holds cached file attributes for at least n seconds.)
- **acregmax**=n (Holds cached file attributes for no more than n seconds.)
- **acdirmin**=n (Holds cached directory attributes for at least n seconds.)
- **acdirmax**=n (Holds cached directory attributes for no more than n seconds.)

PARSEC Group
Our Trainers Consult. Our Consultants Train.

AdvFS Fragmentation

AdvFS file-system fragmentation:

File fragmentation reduces the read/write performance because more I/O operations are required to access a fragmented file.

The **defragment** utility attempts to reduce fregmentation in a file domain by making files more contiguous. Defragmenting a file domain often makes the free space on a disk more contiguous, resulting in less fragmented file allocations in the future.

To check current fragmentation levels on a domain, run “defragment –vn” against the domain(s) in question.

PARSEC Group

Our Trainers Consult. Our Consultants Train.

“defragment” Utility

Example of using the **defragment** command to check fragmentation levels:

```
root@marquis in / --> defragment -vn crash_domain
defragment: Gathering data for domain 'crash_domain'
        Current domain data:
           Extents:                    205
           Files w/extents:            205
           Avg exts per file w/exts:  1.00
           Aggregate I/O perf:        100%
           Free space fragments:       27
                          <100K    <1M    <10M    >10M
           Free space:      0%      0%      0%    100%
           Fragments:        0       7       3      17
```

Extent: A contiguous area of disk space that AdvFS allocates to a file.

When storage is added to a file, it is grouped in extents.

PARSEC Group
Our Trainers Consult. Our Consultants Train.

“advfsstat” Command

Use the “**advfsstat**” command to generate AdvFS performance statistics:

```
root@marquis in / --> advfsstat -i 1 -c 1 crash_domain (1 second increments, 1 iteration)

Domain -1113843636.284579- Stats -
      Deref 2             Refhit     2         Refhitwait 0          R_ahead     0
      Unpin 0             Pinhit     0         Pinhitwait 0          Pinreads    0
      Lazy  0             Log        0         Blocking   0          Clean       0
                          Ubchit     2         Unconsol   0          ConsolAbort 0
                          UnpinMeta 0          UnpinFtx   0          UnpinData   0
                          DerefMeta 2          DerefFtx   2          DerefData   0
Disk     Reads    Writes    Rglobs   AveRglob    Wglobs   AveWglob
----     -----    ------    ------   --------    ------   --------
  1        0        0        0        0        0        0    2        0        0        0        0        0        0
```

(See “man advfsstat” for definitions of the various fields in this output)

PARSEC Group
Our Trainers Consult. Our Consultants Train.

“balance” Command

Use the “**balance**” command to balance the percentage of used space more evenly across multi-volume AdvFS domains:

```
root@marquis in / --> balance -v crash_domain (-v=verbose)
balance: Balancing domain 'crash_domain'
balance: Balanced domain 'crash_domain‘

root@marquis in / --> showfdmn -k crash_domain

              Id              Date Created  LogPgs  Version  Domain Name
4263e7b4.000457a3  Mon Apr 18 10:00:36 2005     512        4  crash_domain

  Vol    1K-Blks       Free      % Used  Cmode  Rblks  Wblks  Vol Name
   1L    4190040    3330464     21%     on    256    256  /dev/disk/dsk4c
   2     4190040    3435736     18%     on    256    256  /dev/disk/dsk5c
          ----------  ----------  ------
          8380080    6766200     19%
```

“migrate” Command

Use the “**migrate**” command to move a file to another volume in the same file domain. Moving a file/files in this manner has two performance advantages:

- Can take advantage of a faster, more robust volume in a file domain.
- By migrating a file/files, discontiguous file extents become more contiguous.

Alternatively, the “**vfast**” **utility (new in Tru64 UNIX Version 5.1B)** performs a number of file-system optimization techniques that:

- Reduce file fragmentation
- Equalize the I/O load
- Balance volume free space

(See the man pages for “migrate” and “vfast” for more details)

PARSEC Group
Our Trainers Consult. Our Consultants Train.

“collect” Utility

The “**collect**” utility is a robust data collection tool that captures detailed statistics regarding a wide variety of Tru64 UNIX performance categories (process utilization, I/O statistics, virtual memory performance, etc.). You can use the collect utility to record all or select categories about the running system.

Example collect output:

```
………..

#### RECORD    1 (1112914690:52) (Thu Apr  7 16:58:10 2005) ####

# Process Statistics (RSS & VSZ in KBytes)
#  PID     User  %CPU  RSS  VSZ  UsrTim SysTim IBk OBk Maj Min         Command
    0    root   0.0  43M 1.4G   0.000  0.000   0   0   0   0      kernel idle
    1    root   0.0  98K 622K   0.000  0.000   0   0   0   0             init
    3    root   0.0 1.0M 1.6M   0.000  0.000   0   0   0   0         kloadsrv

……………..
```

PARSEC Group
Our Trainers Consult. Our Consultants Train.

“collect” Utility

```
# DISK Statistics
#DSK     NAME  B/T/L    R/S RKB/S  W/S WKB/S   AVS   AVW  ACTQ   WTQ  %BSY
   0   cdrom0  0/0/0      0     0    0     0  0.00  0.00  0.00  0.00  0.00
   1     dsk0  2/0/0      0     0    0     0  0.00  0.00  0.00  0.00  0.00
   2     dsk2  2/1/0      0     0    0     0  0.00  0.00  0.00  0.00  0.00
   3     dsk1  2/2/0      0     0    0     0  0.00  0.00  0.00  0.00  0.00

# TAPE Statistics
#NUM     NAME  B/T/L   R/S RKB/S  W/S WKB/S QLEN
   0    tape0  2/3/0     0     0    0     0    0

# CPU SUMMARY
# USER  SYS IDLE WAIT  INTR  SYSC    CS RUNQ  AVG5 AVG30 AVG60 FORK VFORK
     0    0   99    0     2   611   280    0  0.07  0.06  0.03 0.00 0.00
# SINGLE CPU STATISTICS
#   CPU USER  SYS IDLE WAIT
      0    0    0   99    0

# MEMORY STATISTICS
#(<------------ MegaBytes -------------> <--------- Pages/sec -------->)
#  Free   Swap    Act   InAc  Wire   UBC PI PO Zer Re COW SW HIT  PP ALL
    674     70    236      0    71   172 21  0  93  0  22  0   0   0   0
……………………….
```

PARSEC Group
Our Trainers Consult. Our Consultants Train.

“collect” Utility

```
# FileSystem Statistics
# FS                    Filesystem Capacity     Free
   0                 root_domain#root      384      271
   1                           /proc        0        0
   2                  usr_domain#usr     6191     1250
   3                  usr_domain#var     6191     1250

#  Network Statistics
#Cnt         Name  Inpck  InErr Outpck OutErr   Coll    IKB    OKB %BW
   0          lo0      0      0      0      0      0      0     0   0
   1          sl0      0      0      0      0      0      0     0   0
   2          tu0      2      0      0      0      0      0     0   0
   3         tun0      0      0      0      0      0      0     0   0
   4         tun1      0      0      0      0      0      0     0   0

#  Message Queue Statistics
#  ID         Key  OUID  BYTES   Cnt   SPID   RPID     STIME     RTIME     CTIME
    0  1099236080     0      0     0      3    788 1112799054 1112799054 1112799
015

#  TTY Statistics
#     In    Out    Can    Raw
       0     12      0      0
```

PARSEC Group

Our Trainers Consult. Our Consultants Train.

System Tuning/Performance References

Tru64 UNIX System Configuration and Tuning Guide (Tru64 UNIX 5.1B):
http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ ARH9GCTE/TITLE.HTM
(Also note Part 2, Section 4 regarding “Tuning Oracle version 8.1.7.x/9i” on Tru64 UNIX 5.1B)

Tru64 UNIX System Administration Guide (Tru64 UNIX 5.1B):
http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ ARH9FETE/TITLE.HTM

Tru64 UNIX Advanced File System Administration Guide (Tru64 UNIX 5.1B):
http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ ARH96DTE/TITLE.HTM

Tru64 UNIX Network Administration (Connections; Tru64 UNIX 5.1B):
http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ ARH9CDTE/TITLE.HTM

Tru64 UNIX Network Administration (Services; Tru64 UNIX 5.1B):
http://h30097.www3.hp.com/docs/base_doc/DOCUMENTATION/V51B_HTML/ ARPPCBTE/TITLE.HTM

PARSEC Group

Our Trainers Consult. Our Consultants Train.

Find the latest webinars at:

OpenVMS PLANET.org

News, Views, Tips, Forums, Jobs, Resources, and More!

hp Tru64 Performance

Questions?

www.parsec.com | 888-4-PARSEC