

Tru64 issues with Infinite limits

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.tru64/2005-02/0009.html>

From: Chris Heller (*tomaco_at_gmail.com*)

Date: 02/02/05

Date: 2 Feb 2005 06:22:53 -0800

I posted this message on tru64.org, I'm posting it here beacuse I figure the more eyes that see it hte better.

Hello,

Sorry, this post is packed with data, but I hope somebody could explain to me a thing or two about Tru64 limits.

An automated test in my nightly build was failing due to Out of Memory errors. Trying to determine just what was going on, I wrote a simple program[1] to allocate as many 1 megabyte chunks as it can, I set my limits to infinity and tested the program out.

I've posted my results below

```
$ uname -a
OSF1 tru64.site.com V4.0 1530 alpha
$ limit datasize unlimited
$ limit stacksize unlimited
$ limit memoryuse unlimited
$ limit vmemoryuse unlimited
limit: vmemoryuse: Can't remove limit (Invalid argument)
$ limit -h
cputime unlimited
filesize unlimited
datasize 4294967296 kbytes
stacksize 4294967296 kbytes
coredumpsize unlimited
memoryuse 508720 kbytes
vmemoryuse 4294967296 kbytes
descriptors 4096
$ mem-test
block 0 allocated
...
block 120 allocated
swapspace below 10 percent
block 121 allocated
...
```

```
block 145 allocated
Unable to obtain requested swap space
--- previous line repeats 49 more times ---
malloc(): Not enough space
Cannot allocate block 146
$
```

Let me show you some information about the system:

```
$ vmstat -P
```

```
Total Physical Memory = 512.00 M
= 65536 pages
```

Physical Memory Clusters:

```
start_pfn end_pfn type size_pages / size_bytes
0 171 pal 171 / 1.34M
171 65419 os 65248 / 509.75M
65419 65536 pal 117 / 936.00k
```

Physical Memory Use:

```
start_pfn end_pfn type size_pages / size_bytes
171 268 unixtable 97 / 776.00k
268 283 scavenge 15 / 120.00k
283 622 text 339 / 2.65M
622 692 data 70 / 560.00k
692 799 bss 107 / 856.00k
799 805 cfgmngmt 6 / 48.00k
805 807 unixtable 2 / 16.00k
807 820 pmap 13 / 104.00k
820 1847 vmtables 1027 / 8.02M
1847 65419 managed 63572 / 496.66M
=====
```

```
Total Physical Memory Use: 65248 / 509.75M
```

Managed Pages Break Down:

```
free pages = 53585
active pages = 33
inactive pages = 3075
wired pages = 3861
ubc pages = 3034
=====
Total = 63588
```

WIRED Pages Break Down:

```
vm wired pages = 525
ubc wired pages = 0
```

```
meta data pages = 1957
malloc pages = 673
contig pages = 279
user ptepages = 364
kernel ptepages = 54
free ptepages = 8
=====
Total = 3860
```

```
$ swapon -s
Swap partition /dev/rz1b (default swap):
Allocated space: 25088 pages (196MB)
In-use space: 1 pages ( 0%)
Free space: 25087 pages ( 99%)
```

```
Total swap allocation:
Allocated space: 25088 pages (196MB)
Reserved space: 6311 pages ( 25%)
In-use space: 1 pages ( 0%)
Available space: 18777 pages ( 74%)
```

```
$ sysconfigdb -l vm
```

```
vm:
ubc-minpercent = 10
ubc-maxpercent = 100
ubc-borrowpercent = 20
ubc-maxdirtywrites = 5
vm-max-wrpgio-kluster = 32768
vm-max-rdpgio-kluster = 16384
vm-cowfaults = 4
vm-mapentries = 200
vm-maxvas = 1073741824
vm-maxwire = 16777216
vm-heappercnt = 7
vm-vpagemax = 16384
vm-segmentation = 1
vm-ubcpagesteal = 24
vm-ubcdirtypercent = 10
vm-ubcseqstartpercent = 50
vm-ubcseqpercent = 10
vm-csubmapsize = 1048576
vm-ubcbuffers = 256
vm-syncswapbuffers = 128
vm-asyncswapbuffers = 4
vm-clustermap = 1048576
vm-clustersize = 65536
vm-zone_size = 0
vm-kentry_zone_size = 16777216
vm-syswiredpercent = 80
vm-inswappedmin = 1
```

comp.unix.tru64: Tru64 issues with Infinite limits

```
vm-page-free-target = 128
vm-page-free-swap = 74
vm-page-free-hardswap = 2048
vm-page-free-min = 20
vm-page-free-reserved = 10
vm-page-free-optimal = 74
vm-page-prewrite-target = 256
dump-user-pte-pages = 0
kernel-stack-guard-pages = 1
vm-min-kernel-address = 18446744071562067968
malloc-percpu-cache = 1
vm-aggressive-swap = 0
vm_page_color_private = 0
vm-map-index-count = 64
vm-map-index-rebalance = 128
vm-map-index-enabled = 1
vm-map-index-hiwat = 4
vm-map-index-lowat = 2
new-wire-method = 1
vm-segment-cache-max = 50
vm-page-lock-count = 0
gh-chunks = 0
gh-min-seg-size = 8388608
gh-fail-if-no-mem = 1
private-text = 0
private-cache-percent = 0
```

```
$ sysconfigdb -l proc
```

```
proc:
max-proc-per-user = 64
max-threads-per-user = 256
per-proc-stack-size = 4398046511104
max-per-proc-stack-size = 4398046511104
per-proc-data-size = 4398046511104
max-per-proc-data-size = 4398046511104
max-per-proc-address-space = 4398046511104
per-proc-address-space = 4398046511104
executable_stack = 0
autonice = 0
autonice-time = 600
autonice-penalty = 4
open-max-soft = 4096
open-max-hard = 4096
ncallout_alloc_size = 32768
round-robin-switch-rate = 0
round_robin_switch_rate = 0
sched-min-idle = 0
sched_min_idle = 0
give-boost = 1
give_boost = 1
```

```
maxusers = 32
task-max = 277
thread-max = 552
num-wait-queues = 64
num-timeout-hash-queues = 32
enable_extended_uids = 0
enhanced-core-name = 0
enhanced-core-max-versions = 16
```

I ran vmstat while running the test program to see what the effects were, here is the output:

```
# 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
2 57 20 6187 53K 3862 297K 42K 160K 411 33K 0 1 73 53 0
0 100
2 57 20 6202 53K 3862 110 16 67 0 11 0 8 59 63 0
1 99
4 56 20 8247 51K 3871 42 0 41 0 0 0 42 352 151 2
18 80
4 56 20 19K 40K 3919 4083 18 4040 0 15 0 174 849 414 7
93 0
2 57 20 6202 53K 3869 10K 0 10K 0 0 0 104 1K 346 9
56 35
2 57 20 6202 53K 3870 3934 0 3909 0 0 0 5 58 63 0
1 99
3 57 20 14K 45K 3897 42 0 41 0 0 0 129 739 324 6
67 27
3 57 20 24K 35K 3936 9295 18 9252 0 15 0 180 1K 470 10
90 0
2 57 20 6202 53K 3870 9529 0 9529 0 0 0 23 827 154 4
11 85
```

You can see where the program was run (I ran it twice), the 'act' column dips to 19K the first time and 14K,24K the second time.

What am I forgetting? My limits are at ridiculous levels, the system only has 512MB of physical RAM, and I have limits up to the 4GB level. Yet, I cannot make my test allocate more than 145MB of memory, and its hitting swap way before I'd expect it to.

Is there a limit I have not set to infinity that I should? Is there something else entirely that I could look at?

This problem has stumped me, I hope there is a Tru64 administrator out there with the knowledge to get me unstuck.

Regards,
Chris Heller

[1]

Here is the listing of the memory test program:

```
#include <stdio.h>
#include <stdlib.h>

int
main(void)
{
int i;
void *blocks[1024];

for (i = 0; i < 512; i++)
{
void *b = malloc(1048576);
if (b == NULL)
{
perror("malloc()");
printf("Cannot allocate block %d\n", i);
return 0;
}

memset(b, 42, 1048576);

blocks[i] = b;

printf("block %d allocated\n", i);
fflush(stdout);
}

return 0;
}
```