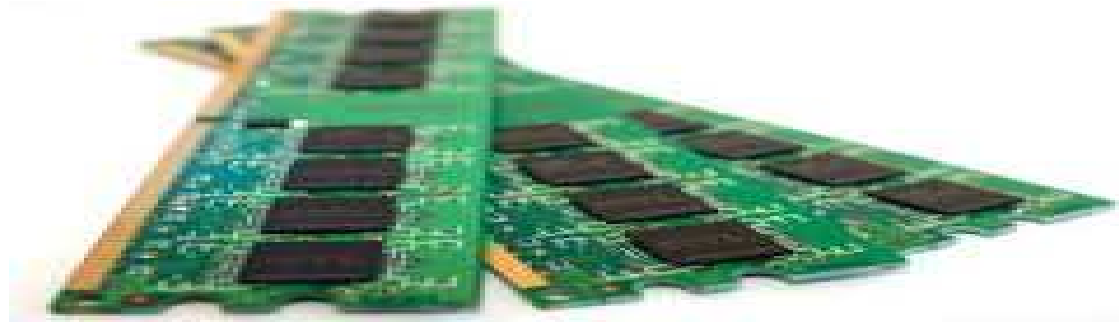


Dude, Where's My Memory?



PUGCHALLENGE EXCHANGE
AMERICAS

Nectarios Daloglou, White Star Software
nd@wss.com

White Star Software

A Few Words about the Speaker

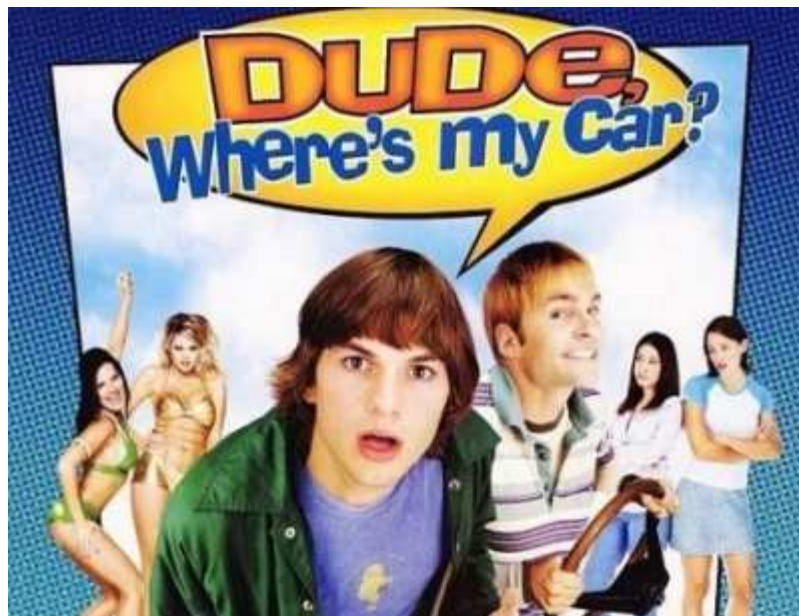


-
- Nectar Daloglou; Progress & QAD since 2000.
 - Performed specialized services at more than 60 Progress customer sites:
 - Progress Database Administration
 - Install/Upgrades/Migrations of Progress and QAD Applications
 - Technical Audits / Performance Tuning
 - Business Continuity Strategies
 - Now part of White Star Software
 - Recently helped migrate 70 Progress environments from AIX to Linux

Dude, Where's my Memory?



- A discussion about memory, not storage
- Focused on Linux
- Sorry, nothing to do with the movie:



White Star Software

Agenda

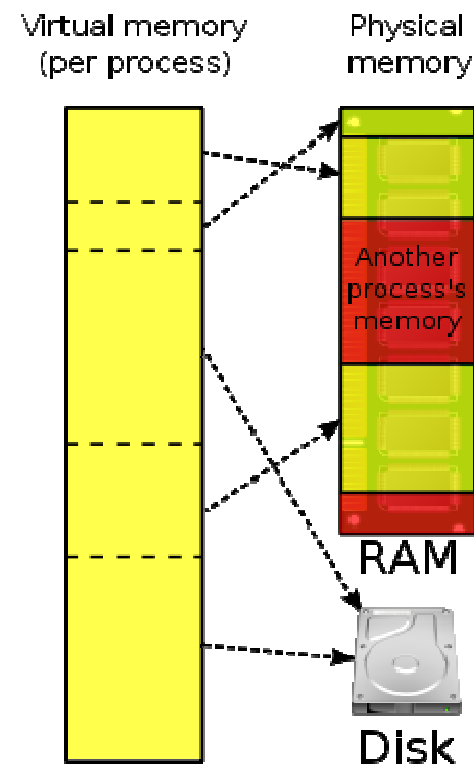


- Memory 101
- Calculating Memory
- Memory Consumers
- Out of Memory Killer
- Out of Memory Scenarios
- Questions



Virtual Memory

Maps virtual addresses into physical addresses



Source: https://en.wikipedia.org/wiki/Virtual_memory

White Star Software



Virtual Memory Characteristics

- Divided into pages:

```
# getconf PAGE_SIZE
```

```
4096
```

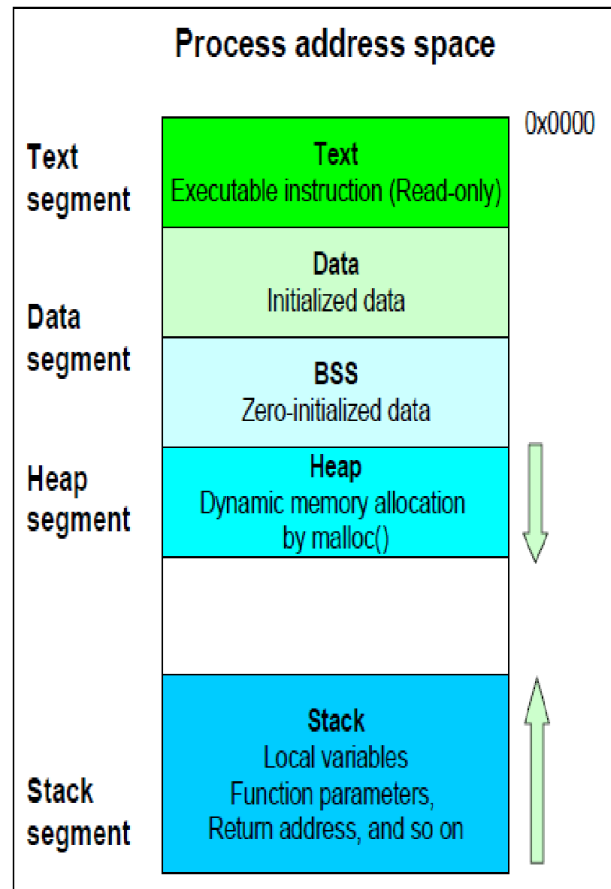
- 32-bit virtual memory space: 2^{32} bytes = 4 GB
- 64-bit virtual memory space: 2^{64} bytes =
 - A lot more space: 18,446,744,073,709,551,616



Virtual Memory Features

- Abstraction of hardware
- Process isolation
- Mapping outside of RAM
- Memory Sharing
- Lazy Allocation

Virtual Memory Segments



Source: http://techblog.cloudperf.net/2016/07/how-linux-kernel-manages-application_18.html



Memory Types

- Private Memory
- Shared Memory
- Anonymous Memory
- File-Backed
- Swap



Memory Types

	PRIVATE	SHARED
ANONYMOUS	1 <pre>1 stack 2 malloc() 3 mmap(ANON, PRIVATE) 4 brk()/sbrk()</pre>	2 <pre>1 mmap(ANON, SHARED)</pre>
FILE-BACKED	3 <pre>1 mmap(fd, PRIVATE) 2 binary/shared libraries</pre>	4 <pre>1 mmap(fd, SHARED)</pre>

Source: <https://techtalk.intersec.com/2013/07/memory-part-1-memory-types/>



Memory Analysis: pmap -x <pid>

- pmap -x <pid>:

```
/data/protop/spawn# pmap -x 3376
3376:  _progres -pf spawn.pf -pf mfgpro.pf -param /tmp/mfgpro.flg
Address          Kbytes      RSS      Dirty Mode      Mapping
0000000000400000    9140      7128         0 r-x--  _progres
000000000104b000    1528       144        144 rwx--  [ anon ]
00007f58c9343000 2354408    7232       7232 rwxS-  [ shmid=0xe0000 ]
00007f58c9343000 2354408         0         0 rwxS-  [ shmid=0xe0000 ]
00007f5958e7d000   32772        16        16 rwx--  [ anon ]
00007f5958e7d000   32772         0         0 rwx--  [ anon ]
00007f595ae7e000     48         48         0 r-x--  libnss_files-2.17.so
00007f595ae7e000     48         0         0 r-x--  libnss_files-2.17.so
00007f595ae8a000    2044         0         0 -----  libnss_files-2.17.so
00007f595ae8a000    2044         0         0 -----  libnss_files-2.17.so
00007f595b089000     4         4         4 r-x--  libnss_files-2.17.so
00007f595b089000     4         0         0 r-x--  libnss_files-2.17.so
00007f595b08a000     4         4         4 rwx--  libnss_files-2.17.so
```

Memory Analysis:

/proc/<PID>/smaps



```
/proc/3376# more smaps
00400000-00ced000 r-xp 00000000 ca:01 525812
/usr/dlcl16/bin/_progres
Size:                9140 kB
Rss:                 7128 kB
Pss:                 99 kB
Shared_Clean:        7128 kB
Shared_Dirty:         0 kB
Private_Clean:        0 kB
Private_Dirty:        0 kB
Referenced:          7128 kB
Anonymous:           0 kB
AnonHugePages:       0 kB
ShmemPmdMapped:      0 kB
Shared_Hugetlb:      0 kB
Private_Hugetlb:     0 kB
Swap:                0 kB
SwapPss:             0 kB
KernelPageSize:      4 kB
MMUPageSize:         4 kB
Locked:              0 kB
VmFlags: rd ex mr mw me dw
```

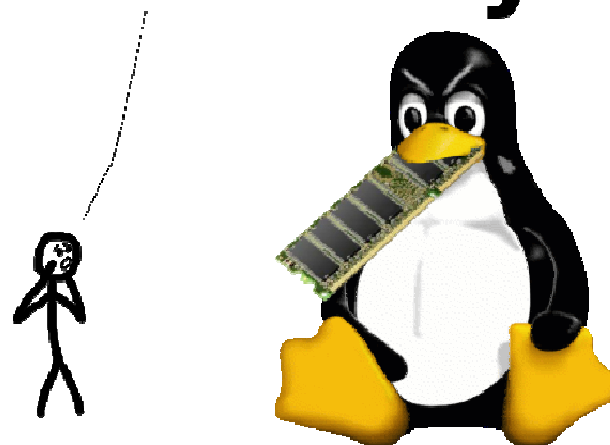


Memory Status

#free -m

	total	used	free	shared	buffers	cached
Mem:	14966	13451	1515	0	234	237
-/+ buffers/cache:		12979	1987			
Swap:	4094	367	3727			

Linux ate my ram!



Don't Panic! Your ram is fine!

Source: <http://www.linuxatemyram.com/>

White Star Software

Swap



- Reserved space to be used as virtual memory
- Read from disk via swap partition or file
- Much slower than RAM
- Stores inactive pages



Swap Usage



- Find which process is using swap: getswap.sh

```
#!/bin/bash
# Get current swap usage for all running processes
# Erik Ljungstrom 27/05/2011
SUM=0
OVERALL=0
for DIR in `find /proc/ -maxdepth 1 -type d | egrep "^/proc/[0-9]"` ; do
    PID=`echo $DIR | cut -d / -f 3`
    PROGNAME=`ps -p $PID -o comm --no-headers`
    for SWAP in `grep Swap $DIR/smaps 2>/dev/null | awk '{ print $2 }'`
    do
        let SUM=$SUM+$SWAP
    done
    echo "PID=$PID - Swap used: $SUM - ($PROGNAME )"
    let OVERALL=$OVERALL+$SUM
    SUM=0
done
echo "Overall swap used: $OVERALL"
```

getswap.sh



```
/data# ./getswap.sh |grep -v "Swap used\ : 0"
```

```
PID=2943 - Swap used: 1336 - (sshd )  
PID=2945 - Swap used: 1376 - (sshd )  
PID=2946 - Swap used: 744 - (bash )  
PID=2969 - Swap used: 1272 - (sudo )  
PID=2970 - Swap used: 800 - (su )  
PID=2971 - Swap used: 848 - (bash )  
PID=2992 - Swap used: 200 - (bash )  
PID=28088 - Swap used: 32656 - (java )  
Overall swap used: 63393
```


Pin Shared Memory (-pinshm)



- Use `-pinshm` to ensure database shared memory does not end up in swap
- Not available on AIX or Windows



Adjust “Swappiness”



- Swappiness controls the relative weight given to swapping out runtime memory
- Adjust in `/proc/sys/vm/swappiness` or `sysctl -w vm.swappiness=##`

Value	Strategy
<code>vm.swappiness = 0</code>	The kernel will swap only to avoid an out of memory condition, when free memory will be below <code>vm.min_free_kbytes</code> limit. See the "VM Sysctl documentation" .
<code>vm.swappiness = 1</code>	Kernel version 3.5 and over, as well as Red Hat kernel version 2.6.32-303 and over: Minimum amount of swapping without disabling it entirely.
<code>vm.swappiness = 10</code>	This value is sometimes recommended to improve performance when sufficient memory exists in a system.
<code>vm.swappiness = 60</code>	The default value.
<code>vm.swappiness = 100</code>	The kernel will swap aggressively.



Monitor Swap

- Use `vmstat` to check for frequent swapping

```
#vmstat 5 10
procs
r  b  w   swpd   free  buff  cache  si  so  bi  bo  in   cs  us  sy  id
.  .  .
1  0  0  13344  1444  1308  19692   0 168 129  42 1505   713 20  11  69
1  0  0  13856  1640  1308  18524  64 516 379 129 4341   646 24  34  42
3  0  0  13856  1084  1308  18316  56  64  14   0  320  1022 84   9   8
```

Agenda



- Memory 101
- Calculating Memory
- Memory Consumers
- Out of Memory Killer
- Out of Memory Scenarios
- Questions



Calculating with top

```
#top
  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 28088 root        20   0 3357m   28m 8148  S   0.0   0.4    1:59.48  java
   3390 root        20   0 2608m  302m   88m  R   2.6   3.8   19:47.58  _progres
   3376 root        20   0 2432m   25m   19m  R   2.6   0.3   19:49.51  _progres
 19946 root        20   0 2400m   95m   88m  R   2.3   1.2   56:31.13  _progres
 19947 root        20   0 2400m   94m   88m  R   2.6   1.2   56:32.26  _progres
 19948 root        20   0 2400m   94m   88m  R   3.0   1.2   56:31.78  _progres
 19949 root        20   0 2400m   94m   88m  R   2.6   1.2   56:31.38  _progres
 19950 root        20   0 2400m   94m   88m  R   2.6   1.2   56:30.52  _progres
 19951 root        20   0 2400m   94m   87m  R   3.0   1.2   56:31.57  _progres
 19916 root        20   0 2322m  2.2g  2.2g  S   0.4  28.7    0:28.29  _mprosrv
```

Issues:

- VIRT (Virtual Memory) not actual Physical Memory
- RES (Resident Memory) actual Physical Memory but also includes shared memory
- SHR is a subset of shared memory that is file-backed

Calculating with ps



Similar Issues:

```
/data# ps aux |more
USER      PID  %CPU  %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      3376  2.7   0.3  2491044 26536 pts/0    S    Jun02  20:00  _progres
root      3390  2.7   3.7  2671460 310200 pts/0    D    Jun02  19:58  _progres
root     19916  0.0  28.7  2377768 2349960 pts/0    S    Jun01   0:28  _mprosrv
root     19946  2.8   1.1  2458312  97332 pts/0    R    Jun01  56:42  _progres
root     19947  2.8   1.1  2458312  96660 pts/0    R    Jun01  56:43  _progres
root     19948  2.8   1.1  2458312  96716 pts/0    R    Jun01  56:42  _progres
root     19949  2.8   1.1  2458312  96724 pts/0    R    Jun01  56:42  _progres
root     19950  2.8   1.1  2458312  96768 pts/0    R    Jun01  56:41  _progres
root     19951  2.8   1.1  2458312  96628 pts/0    R    Jun01  56:42  _progres
root     19952  2.8   1.1  2458312  96840 pts/0    R    Jun01  56:42  _progres
root     19953  2.8   1.1  2458312  96760 pts/0    R    Jun01  56:40  _progres
root     19954  2.8   1.1  2458312  96676 pts/0    R    Jun01  56:43  _progres
root     19955  2.8   1.1  2458312  96764 pts/0    R    Jun01  56:42  _progres
root     19956  2.8   1.1  2458312  96728 pts/0    R    Jun01  56:39  _progres
root     19957  2.8   1.1  2458312  96788 pts/0    R    Jun01  56:39  _progres
root     19958  2.8   1.1  2458312  96608 pts/0    R    Jun01  56:41  _progres
root     19959  2.8   0.3  2458312  26476 pts/0    R    Jun01  56:41  _progres
root     28088  0.0   0.3  3438184  29592 ?        Ssl  May31   1:59  java
```

Calculating with smaps extraction



Memuse.sh script:

```
#!/bin/bash
# http://stackoverflow.com/questions/3853655/in-linux-how-to-tell-how-much-
# memory-processes-are-using
MYPID=$1
export MYPID
echo "=====";
echo PID:$MYPID
echo "-----"
Rss=`echo 0 $(cat /proc/$MYPID/smaps | grep Rss | awk '{print $2}' | sed
's#^#+#') | bc;`
Shared=`echo 0 $(cat /proc/$MYPID/smaps | grep Shared | awk '{print $2}' |
sed 's#^#+#') | bc;`
Private=`echo 0 $(cat /proc/$MYPID/smaps | grep Private | awk '{print $2}' |
sed 's#^#+#') | bc;`
Swap=`echo 0 $(cat /proc/$MYPID/smaps | grep Swap | awk '{print $2}' | sed
's#^#+#') | bc;`
Pss=`echo 0 $(cat /proc/$MYPID/smaps | grep Pss | awk '{print $2}' | sed
's#^#+#') | bc;`
Mem=`echo "$Rss + $Shared + $Private + $Swap + $Pss"|bc -l`
```

Calculating with smaps extraction



Sample Output:

```
_progres:                _mprosrv:                java:

=====                 =====                 =====
PID:20015                PID:19916                PID:28088
-----                 -----                 -----
Rss   97092               Rss  2350056             Rss   31396
Shared 90560              Shared 97396              Shared 1400
Private 6532              Private 2252660           Private 29996
Swap   0                  Swap   0                  Swap  31904
Pss   7872                Pss  2262150             Pss   45973
=====                 =====                 =====
Mem   202056              Mem   6962262            Mem   140669
=====                 =====                 =====
```

PSS: Proportional Set Size which is RSS adjusted for sharing



Calculating with pmap -x

1. Add up all [stack] & [anon] resident memory mappings

Such as:

Address	Kbytes	RSS	Dirty	Mode	Mapping
00007ffaf4022000	20	20	20	rwX--	[anon]
00007ffef9b57000	132	44	44	rwX--	[stack]

2. Count shared memory mappings only once:

Address	Kbytes	RSS	Dirty	Mode	Mapping
00007ffa631a4000	2354408	2344460	2344460	rwXs-	[shmid=0xe0000]



Agenda

- Memory 101
- Calculating Memory
- Memory Consumers
- Out of Memory Killer
- Out of Memory Scenarios
- Questions

Broker Parameters Can Affect Memory



- -B (Database Buffers): Measured in DB block size ex.: -B 1,000,000 on 4KB = Apx. 4GB
- -L (Lock-table Entries): 64 bytes / record lock

– Personally calculated 162 bytes in 11.6:

```
data# proserve sports2000 -L 10000000
```

```
05:05:33 BROKER      The startup of this
database requires 1564Mb of shared memory.
Maximum segment size is 2048Mb.
```

- -c (Index Cursors): 84 bytes each

Source: https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/gsdbe/openedge-specific-memory-estimates.html

Broker Parameters Can Affect Memory



- -Mn (Remote Client Servers): 3MB-5MB each
- -n (Number of Users): 2KB each

– Personally calculated 40KB each in 11.6:

```
/data# proserve sports2000 -n 1000
```

```
OpenEdge Release 11.6 as of Fri Oct 16  
18:22:20 EDT 2015
```

```
05:14:28 BROKER          The startup of this  
database requires 44Mb of shared memory.  
Maximum segment size is 1024Mb.
```

Source: https://documentation.progress.com/output/ua/OpenEdge_latest/index.html#page/gsdbe/openedge-specific-memory-estimates.html

Be careful with tablerange/indexrange size

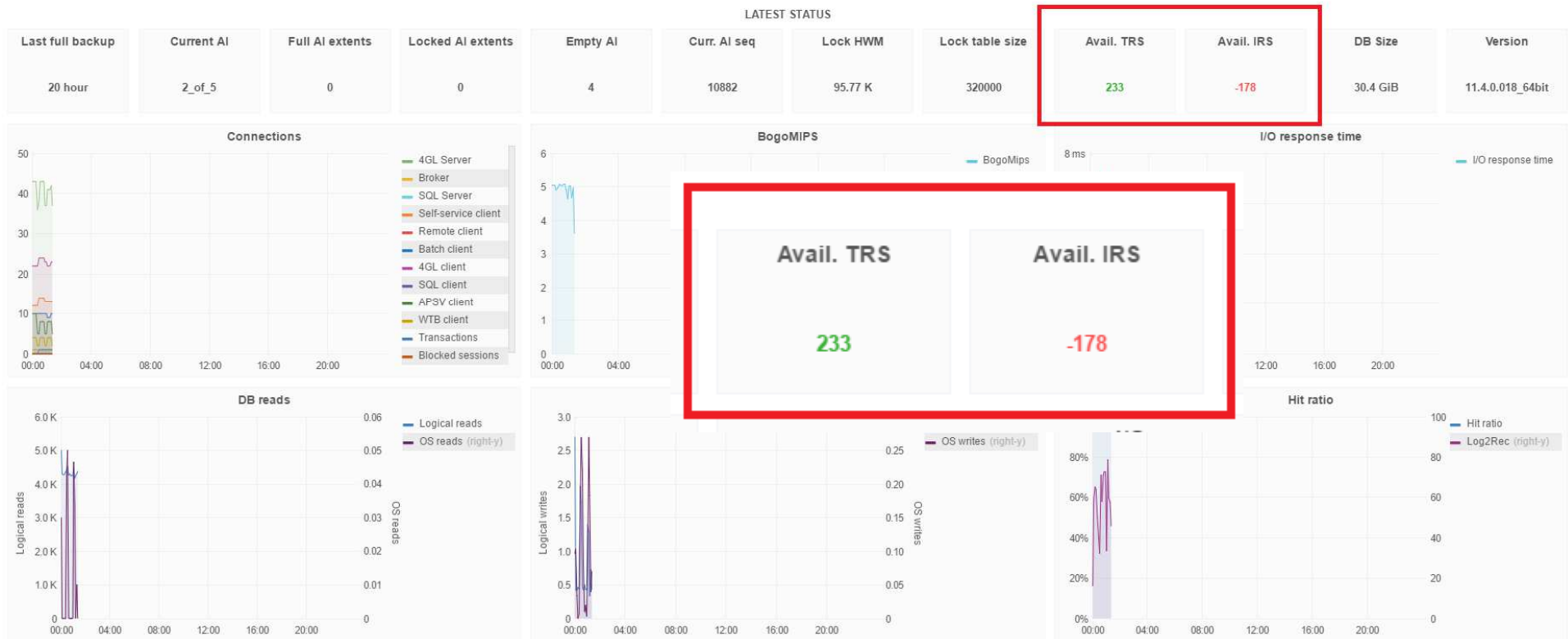


- Adding large values to `-tablerangesize` & `-indexrangesize` can further increase memory usage by a factor of `-n`:

```
/data# proserve sports2000 -n 1000 -tablerangesize 10000  
-indexrangesize 10000
```

```
OpenEdge Release 11.6 as of Fri Oct 16 18:22:20 EDT 2015  
05:24:10 BROKER      The startup of this database requires  
885Mb of shared memory.  Maximum segment size is 1024Mb.
```

Monitor tablerange/indexrange size with ProTop



- Free download: <http://protop.wss.com>

White Star Software

Client Parameters Can Affect Memory



- -Bt (Temporary Table Buffers): n X -tmpbsize size
 - A value of –Bt 50,000 adds a 200MB anonymous page in resident memory:

Address	Kbytes	RSS	Dirty	Mode	Mapping
00007f4fc59a7000	215668	214492	214492	rwx--	[anon]

- Dynamically allocates more memory:
 - -mmax (Maximum Memory)
 - -D (Directory Size)
 - -l (Local Buffer Size)
 - -nb (Nested Blocks)
 - These can be limited by setting the –hardlimit parameter



Agenda

- Memory 101
- Calculating Memory
- Memory Consumers
- Out of Memory Killer
- Out of Memory Scenarios
- Questions



Linux Out of Memory Killer

- A self-preservation mechanism that terminates a process when memory is over committed.
- OOM will kill the most memory consuming process

`/var/log/messages:`

```
Oct 22 16:05:48 s619784dc3v142 kernel: Out of memory: Kill process 5995 (_mprosrv) score 421 or sacrifice child
```

```
Oct 22 16:05:48 s619784dc3v142 kernel: Killed process 5995, UID 65535, (_mprosrv) total-vm:34767444kB, anon-rss:12680kB, file-rss:31617936kB
```

- Likelihood is based on “badness” score



Linux Out of Memory Killer

- Check a process's likelihood of being terminated in `/proc/<pid>/oom_score`:

```
# cat /proc/8224/oom_score
2
```

- Find the process most likely to be killed:

```
# dstat --top-oom
--out-of-memory---
      kill score
_mprosrv          247
```

- Can control likelihood by setting `oom_adj` (Valid range -16 to +15; -17 to exempt:
 - Example: `echo -17 > /proc/5995/oom_adj`



Agenda

- Memory 101
- Calculating Memory
- Memory Consumers
- Out of Memory Killer
- Out of Memory Scenarios
- Questions

Case of the missing memory



- Customer reported OOM crashes
- RedHat recommended additional RAM & additional Swap
- Crashing stopped, but memory was still missing and still swapping:

```
# free -m
```

	total	used	free	shared	buffers	cached
Mem:	40076	39224	851	19961	3	891
-/+ buffers/cache:		38329	1747			
Swap:	65151	4083	61068			

Case of the missing memory



- Calculating memory usage using pmap added up to 20GB vs 42GB being used
- Backup script was writing to a device that did not exist:

```
tar cvf /dev/st0 $DIRS
```

- Memory was being written into memory-backed file system



VMWare Ballooning

- VMWare ESX may reclaim memory from guest
 - Done through a private channel
 - Can cause kernel to swap
- Check for ballooning activity; look for non-zero values in `/sys/kernel/debug/vmmemctl`:

```
# cat /sys/kernel/debug/vmmemctl
target:                0 pages
current:               0 pages
```



Infinite (-l) increases

- Process eventually consumed all the memory

```
[2014/03/17@12:02:46.754-0400] P-8573096 T-1 I ABL 140: (5408) WARNING: -l exceeded. Automatically increasing from 3323700 to 3323710.  
[2014/03/17@12:02:46.760-0400] P-8573096 T-1 I ABL 140: (5408) WARNING: -l exceeded. Automatically increasing from 3323710 to 3323720.  
[2014/03/17@12:02:46.767-0400] P-8573096 T-1 I ABL 140: (5408) WARNING: -l exceeded. Automatically increasing from 3323720 to 3323730.  
[2014/03/17@12:02:46.773-0400] P-8573096 T-1 I ABL 140: (5408) WARNING: -l exceeded. Automatically increasing from 3323730 to 3323740.
```

- Monitor and consider -hardlimit



Conclusion

- Determining actual memory usage is not obvious, separate shared memory and count once
- Monitor to help prevent an OOM failure

Questions?

White Star Software

Questions



-
- Questions or comments? Feel free to e-mail me:

Nectar Daloglou: nd@wss.com

Thank You!

Try ProTop Free



The #1 FREE OpenEdge Database Monitoring Tool



protop

<http://protop.wss.com/>

White Star Software

White Star Software



-
- The oldest and most respected independent DBA consulting firm in the world
 - Five of the world's top OpenEdge DBAs
 - Author of ProTop, the #1 FREE OpenEdge Database Monitoring Tool
 - <http://protop.wss.com/>