Dude, Where’s My Memory?

Nectarios Daloglou, White Star Software

nd@wss.com

EMEA PUG Challenge

Dublin 2018
A Few Words about the Speaker

• Nectar Daloglou; Progress & QAD since 2000.
• Performed specialized services at more than 70 OpenEdge customer sites:
  • Progress Database Administration
  • Install/Upgrades/Migrations of Progress and QAD Applications
  • Technical Audits / Performance Tuning
  • Business Continuity Strategies
Dude, Where’s my Memory?

• A discussion about memory, not storage
• Focused on Linux
• Goal: Basic understanding on memory and how OpenEdge uses it
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 Features

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

Memory Types

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

<table>
<thead>
<tr>
<th></th>
<th>PRIVATE</th>
<th>SHARE</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ANONYMOUS</strong></td>
<td>1</td>
<td>2</td>
<td><strong>PRIVATE</strong></td>
<td><strong>PRIVATE</strong></td>
<td><strong>PRIVATE</strong></td>
</tr>
<tr>
<td>stack</td>
<td></td>
<td></td>
<td>malloc()</td>
<td>malloc()</td>
<td>malloc()</td>
</tr>
<tr>
<td>malloc(ANON, PRIVATE)</td>
<td></td>
<td></td>
<td>brk()/sbrk()</td>
<td>brk()/sbrk()</td>
<td>brk()/sbrk()</td>
</tr>
<tr>
<td><strong>FILE-BACKED</strong></td>
<td>3</td>
<td>4</td>
<td><strong>PRIVATE</strong></td>
<td><strong>PRIVATE</strong></td>
<td><strong>PRIVATE</strong></td>
</tr>
<tr>
<td>mmap(fd, PRIVATE)</td>
<td></td>
<td></td>
<td>binary/shared libraries</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mmap(fd, SHARED)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: [https://techtalk.intersec.com/2013/07/memory-part-1-memory-types/](https://techtalk.intersec.com/2013/07/memory-part-1-memory-types/)
Memory Analysis: `pmap -x <pid>`

```bash
/data/protop/spawn# pmap -x 3376
3376:  _progres -pf spawn.pf -pf mfgpro.pf -param /tmp/mfgpro.flg

<table>
<thead>
<tr>
<th>Address</th>
<th>Kbytes</th>
<th>RSS</th>
<th>Dirty</th>
<th>Mode</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000000000400000</td>
<td>9140</td>
<td>7128</td>
<td>0</td>
<td>r-x--</td>
<td>_progres</td>
</tr>
<tr>
<td>000000000104b000</td>
<td>1528</td>
<td>144</td>
<td>144</td>
<td>rwx--</td>
<td>[ anon ]</td>
</tr>
<tr>
<td>00007f58c9343000</td>
<td>2354408</td>
<td>7232</td>
<td>7232</td>
<td>rwxs-</td>
<td>[ shmid=0xe0000 ]</td>
</tr>
<tr>
<td>00007f58c9343000</td>
<td>2354408</td>
<td>0</td>
<td>0</td>
<td>rwxs-</td>
<td>[ shmid=0xe0000 ]</td>
</tr>
<tr>
<td>00007f5958e7d000</td>
<td>32772</td>
<td>16</td>
<td>16</td>
<td>rwx--</td>
<td>[ anon ]</td>
</tr>
<tr>
<td>00007f5958e7d000</td>
<td>32772</td>
<td>0</td>
<td>0</td>
<td>rwx--</td>
<td>[ anon ]</td>
</tr>
<tr>
<td>00007f595ae7e000</td>
<td>48</td>
<td>48</td>
<td>0</td>
<td>r-x--</td>
<td>libnss_files-2.17.so</td>
</tr>
<tr>
<td>00007f595ae7e000</td>
<td>48</td>
<td>0</td>
<td>0</td>
<td>r-x--</td>
<td>libnss_files-2.17.so</td>
</tr>
<tr>
<td>00007f595ae8a000</td>
<td>2044</td>
<td>0</td>
<td>0</td>
<td>-----</td>
<td>libnss_files-2.17.so</td>
</tr>
<tr>
<td>00007f595ae8a000</td>
<td>2044</td>
<td>0</td>
<td>0</td>
<td>-----</td>
<td>libnss_files-2.17.so</td>
</tr>
<tr>
<td>00007f595b089000</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>r-x--</td>
<td>libnss_files-2.17.so</td>
</tr>
<tr>
<td>00007f595b089000</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>r-x--</td>
<td>libnss_files-2.17.so</td>
</tr>
<tr>
<td>00007f595b08a000</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>rwx--</td>
<td>libnss_files-2.17.so</td>
</tr>
</tbody>
</table>
```
Memory Analysis:
/proc/<PID>/smaps

/proc/3376# more smaps
00400000-00ced000 r-xp 00000000 ca:01 525812
/usr/dlc116/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

<table>
<thead>
<tr>
<th></th>
<th>total</th>
<th>used</th>
<th>free</th>
<th>shared</th>
<th>buffers</th>
<th>cached</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mem:</td>
<td>14966</td>
<td>13451</td>
<td>1515</td>
<td>0</td>
<td>234</td>
<td>237</td>
</tr>
<tr>
<td>+/- buffers/cache:</td>
<td>12979</td>
<td>1987</td>
<td>3727</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Swap:</td>
<td>4094</td>
<td>367</td>
<td>367</td>
<td>0</td>
<td>234</td>
<td>237</td>
</tr>
</tbody>
</table>

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 “Swapiness”

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>vm.swappiness = 0</td>
<td>The kernel will swap only to avoid an out of memory condition, when free memory will be below vm.min_free_kbytes limit. See the &quot;VM Sysctl documentation&quot;.</td>
</tr>
<tr>
<td>vm.swappiness = 1</td>
<td>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.</td>
</tr>
<tr>
<td>vm.swappiness = 10</td>
<td>This value is sometimes recommended to improve performance when sufficient memory exists in a system.</td>
</tr>
<tr>
<td>vm.swappiness = 60</td>
<td>The default value.</td>
</tr>
<tr>
<td>vm.swappiness = 100</td>
<td>The kernel will swap aggressively.</td>
</tr>
</tbody>
</table>

Source: https://en.wikipedia.org/wiki/Swappiness
Monitor Swap

- Use `vmstat` to check for frequent swapping

```bash
#vmstat 5 10
procs memory swap io system cpu
  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

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

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

<table>
<thead>
<tr>
<th>USER</th>
<th>PID</th>
<th>%CPU</th>
<th>%MEM</th>
<th>VSZ</th>
<th>RSS</th>
<th>TTY</th>
<th>STAT</th>
<th>START</th>
<th>TIME</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>3376</td>
<td>2.7</td>
<td>0.3</td>
<td>2491044</td>
<td>26536</td>
<td>pts/0</td>
<td>S</td>
<td>Jun02</td>
<td>20:00</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>3390</td>
<td>2.7</td>
<td>3.7</td>
<td>2671460</td>
<td>310200</td>
<td>pts/0</td>
<td>D</td>
<td>Jun02</td>
<td>19:58</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19916</td>
<td>0.0</td>
<td>28.7</td>
<td>2377768</td>
<td>2349960</td>
<td>pts/0</td>
<td>S</td>
<td>Jun01</td>
<td>0:28</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19946</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>97332</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:42</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19947</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96660</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:43</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19948</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96716</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:42</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19949</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96724</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:42</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19950</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96668</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:41</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19951</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96628</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:42</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19952</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96840</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:42</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19953</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96760</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:40</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19954</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96676</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:43</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19955</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96764</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:42</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19956</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96728</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:39</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19957</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96788</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:39</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19958</td>
<td>2.8</td>
<td>1.1</td>
<td>2458312</td>
<td>96608</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:41</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>19959</td>
<td>2.8</td>
<td>0.3</td>
<td>2458312</td>
<td>26476</td>
<td>pts/0</td>
<td>R</td>
<td>Jun01</td>
<td>56:41</td>
<td>_progres</td>
</tr>
<tr>
<td>root</td>
<td>28088</td>
<td>0.0</td>
<td>0.3</td>
<td>3438184</td>
<td>29592</td>
<td>?</td>
<td>Ssl</td>
<td>May31</td>
<td>1:59</td>
<td>java</td>
</tr>
</tbody>
</table>
```
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:

=======
PID:20015
--------
Rss 97092
Shared 90560
Private 6532
Swap 0
Pss 7872
=================
Mem 202056
=================

_mprosrv:

=======
PID:19916
--------
Rss 2350056
Shared 97396
Private 2252660
Swap 0
Pss 2262150
=================
Mem 6962262
=================

java:

=======
PID:28088
--------
Rss 31396
Shared 1400
Private 29996
Swap 31904
Pss 45973
=================
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:

<table>
<thead>
<tr>
<th>Address</th>
<th>Kbytes</th>
<th>RSS</th>
<th>Dirty</th>
<th>Mode</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>00007ffaf4022000</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>rwx--</td>
<td>[ anon ]</td>
</tr>
<tr>
<td>00007ffef9b57000</td>
<td>132</td>
<td>44</td>
<td>44</td>
<td>rwx--</td>
<td>[ stack ]</td>
</tr>
</tbody>
</table>

2. Count shared memory mappings only once:

<table>
<thead>
<tr>
<th>Address</th>
<th>Kbytes</th>
<th>RSS</th>
<th>Dirty</th>
<th>Mode</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>00007ffa631a4000</td>
<td>2354408</td>
<td>2344460</td>
<td>2344460</td>
<td>rwxs-</td>
<td>[ shmid=0xe0000 ]</td>
</tr>
</tbody>
</table>
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/gsdbi/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:

<table>
<thead>
<tr>
<th>Address</th>
<th>Kbytes</th>
<th>RSS</th>
<th>Dirty Mode</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>00007f4fc59a7000</td>
<td>215668</td>
<td>214492</td>
<td>214492</td>
<td>rwx--</td>
</tr>
</tbody>
</table>

• 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:

```bash
# free -m

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

White Star Software
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 increasing from 3323700 to 3323710.
[2014/03/17@12:02:46.760-0400] P-8573096 increasing from 3323710 to 3323720.
[2014/03/17@12:02:46.767-0400] P-8573096 increasing from 3323720 to 3323730.
[2014/03/17@12:02:46.773-0400] P-8573096 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?
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

http://wss.com/protop

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://wss.com/protop](http://wss.com/protop)