IBM

Technical Forum & Executive Briefing

17 al 21 Octubre 2011

Imagine PODER Imagine CAPACIDAD

Session title – Analyzing Memory Performance

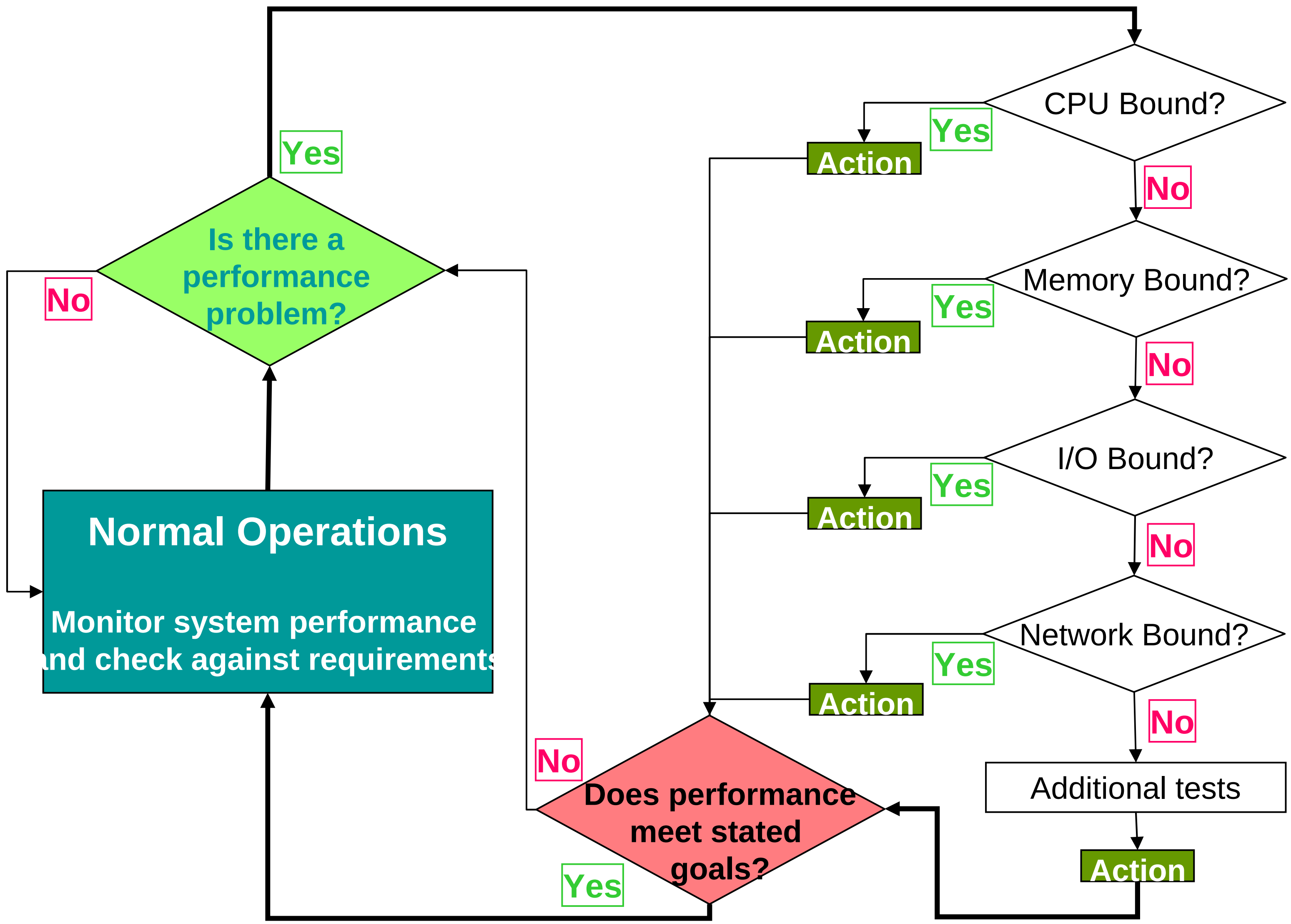
Speaker name – Scott "Tex" Nance

nancet@us.ibm.com

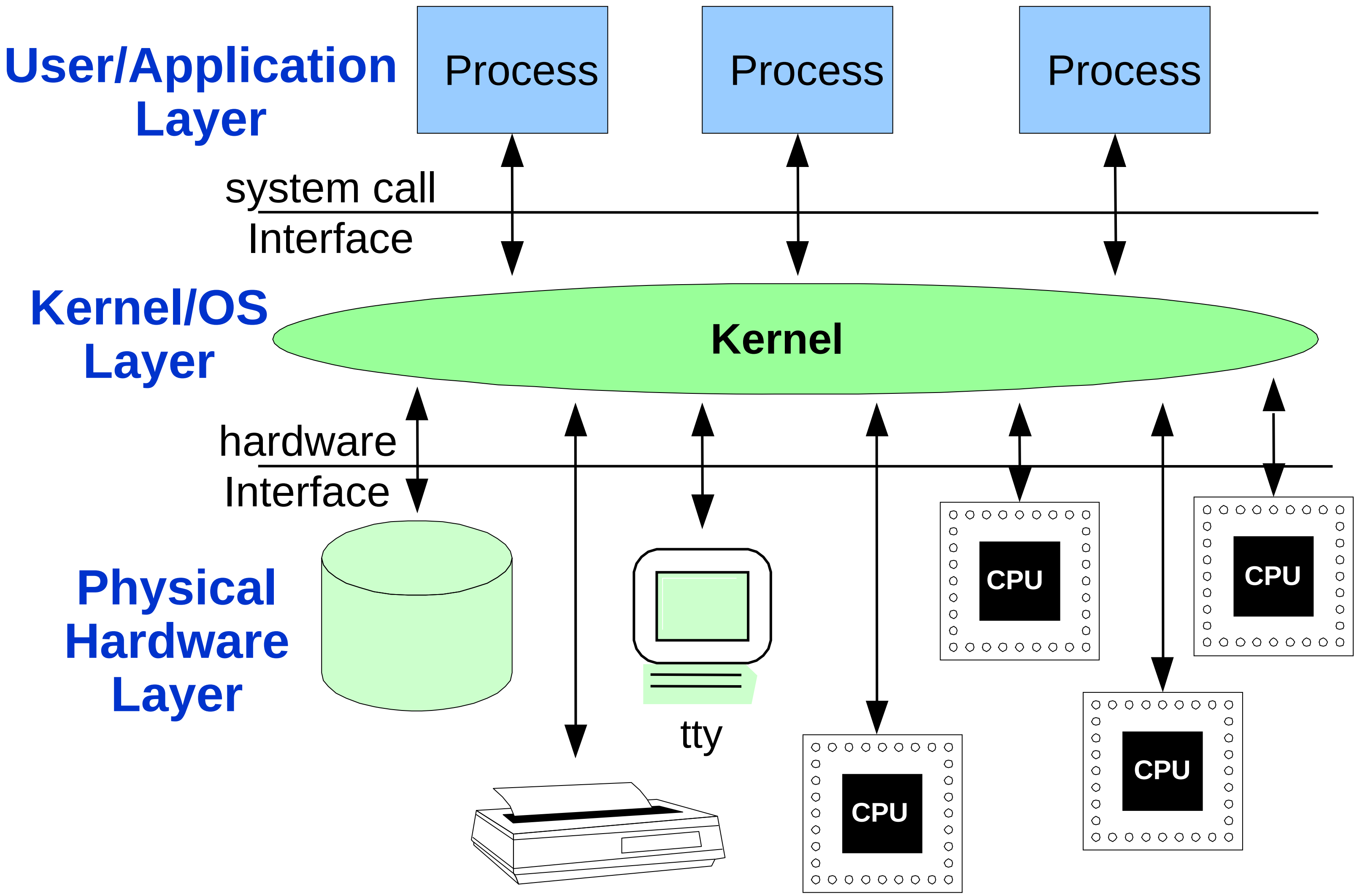
Session Objectives

- Define virtual memory concepts and terminology and explain their impact on memory based performance issues
- Calculate and categorize the memory in use on the system
- Identify which processes are using the most memory
- Determine if a system has enough memory

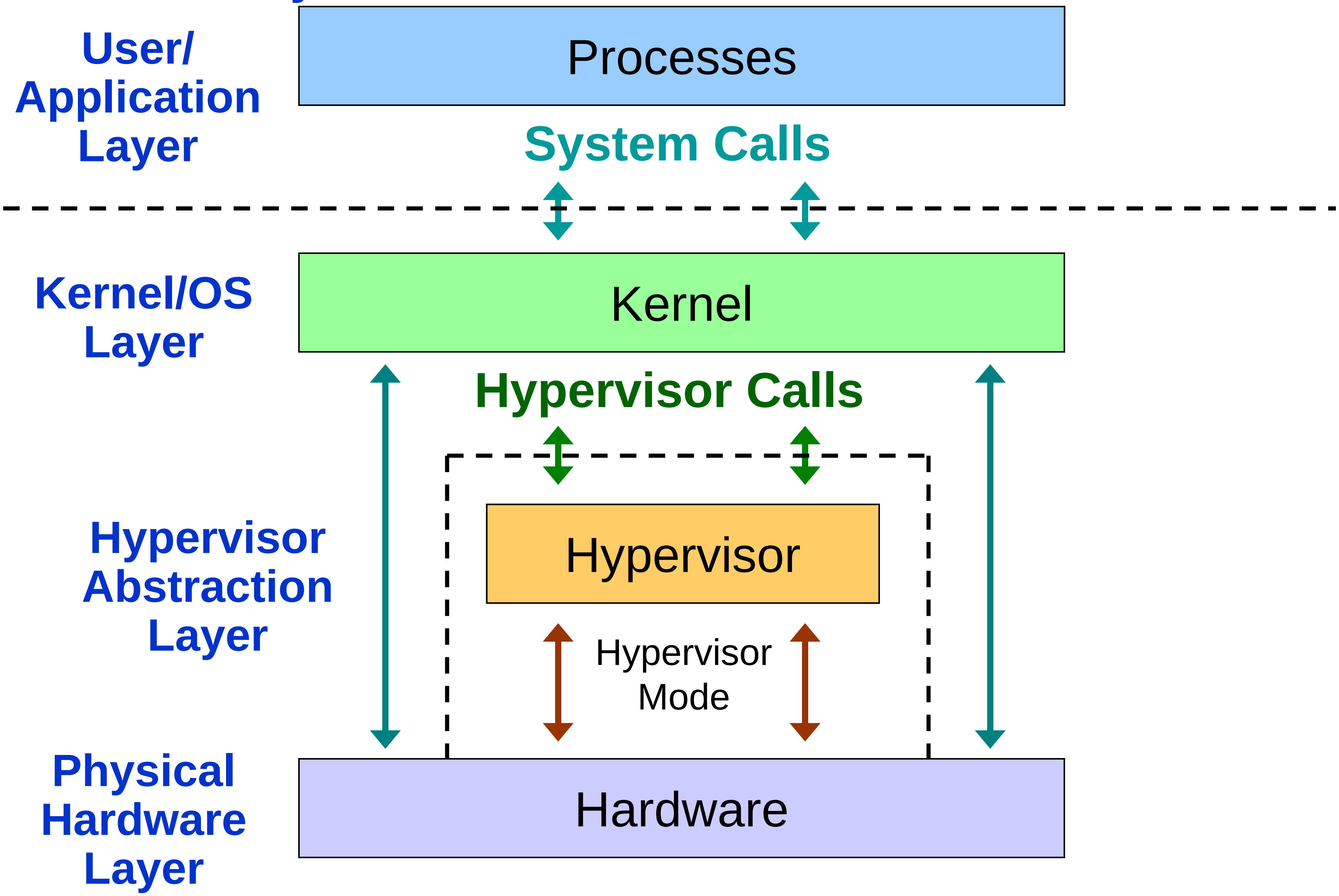
Performance Analysis Flowchart



Traditional System Architecture



Partitioned System Architecture



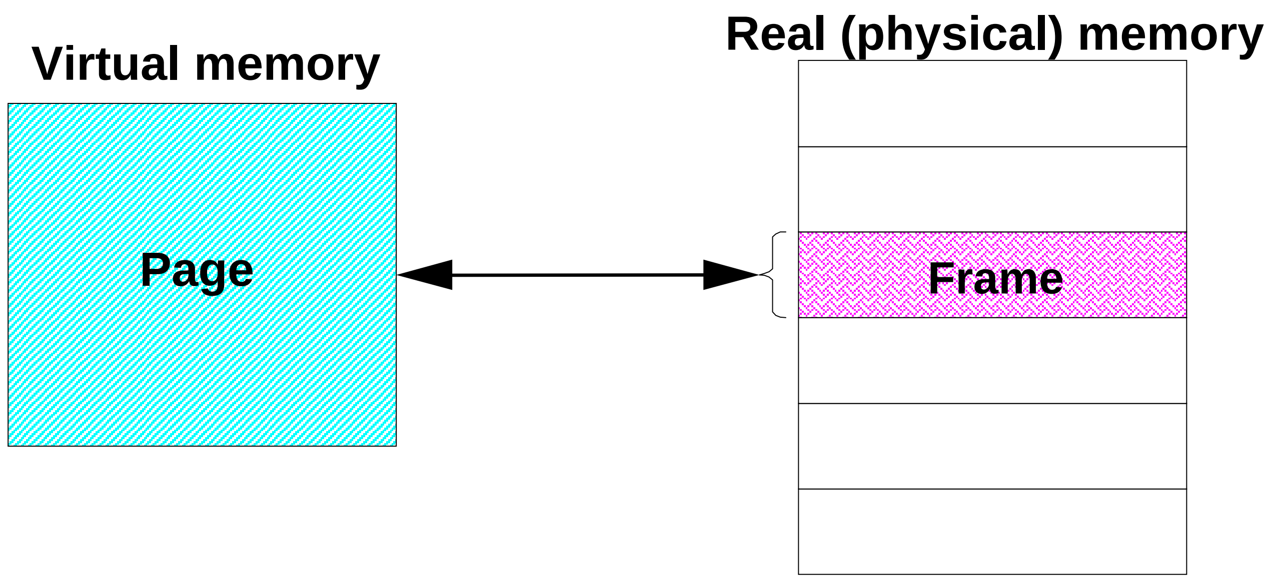
What is the main goal of Memory Tuning?

```
# vmstat -I 5

System configuration: lcpu=4 mem=1024MB

 kthr      memory                  page                     faults              cpu
--------  -----------  ---------------------------   ----------------- ---------------
r  b  p   avm    fre    fi    fo  pi  po   fr    sr   in     sy    cs    us  sy  id  wa
1  0  0 187052  3219  11195   0   0   0  8720  9165  521 87646  7632    8  21  58  12
1  0  0 187067  3214  4332    0   0   0  2697  2697  455 63884  4932    9  18  61  12
1  0  0 187084  3144  3730    0   0   0  2374  2374  389 62610  5618   10  20  60  11
2  0  0 187069  3006  4283    0   0   0  2634  2634  430 65111  6241   10  21  59  11
1  0  0 187213  3048  5145    0   0   0  3979  75936 385 67500  4276    9  23  56  12
0  1  0 187200  3140  14301   0   0   0  13494 13935 428 78735  3471    6  20  60  14
1  0  0 187230  3188  13208   0   0   0  11605 11748 346 126253 12208   8  24  57  11
1  0  0 187376  3135  3070    0   0   0  1092  1188  427 162036 29224  16  30  51   3
1  0  0 187332  3618  4756    0   0   0  3390  3865  414 152360 21478  13  27  54   5
1  0  0 187520  3244  4776    0   0   0  2351  2364  445 162840 30134  13  27  55   5
```

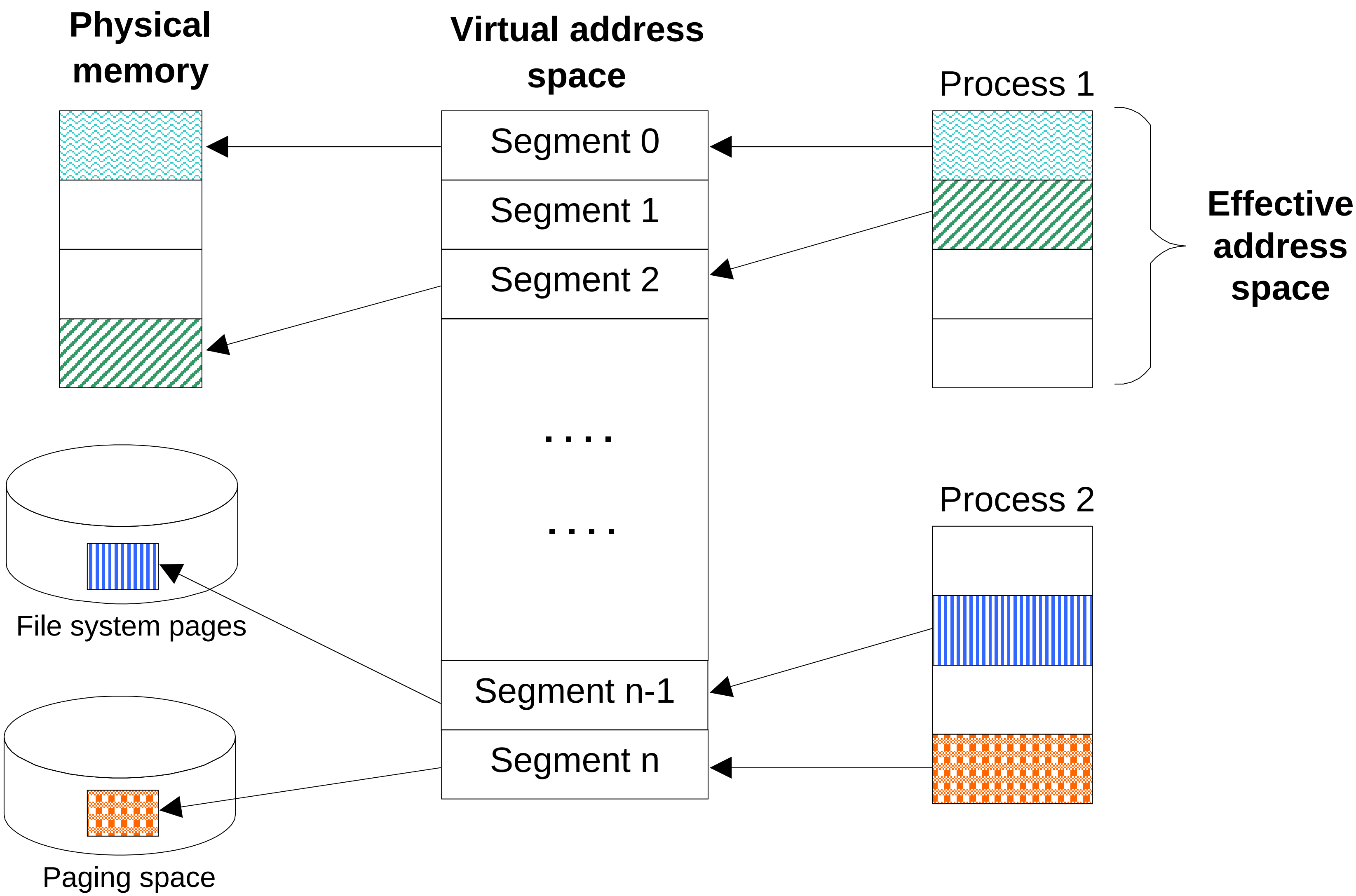
Pages and Frames

- Page size traditionally has been 4096 bytes

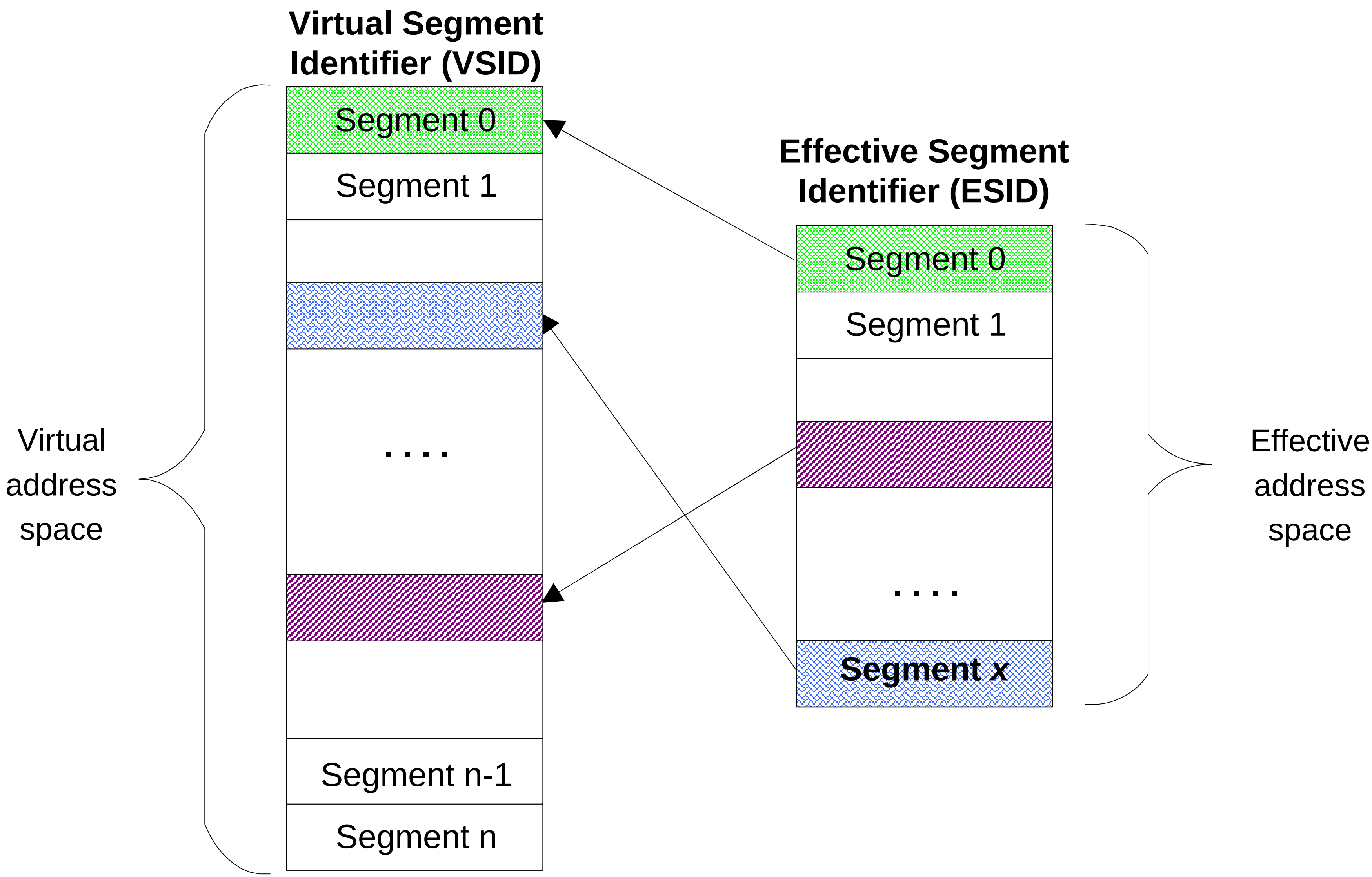
- Newer systems support multiple page sizes
  – Possible sizes are 4 KB, 64 KB, 16 MB and 16 GB



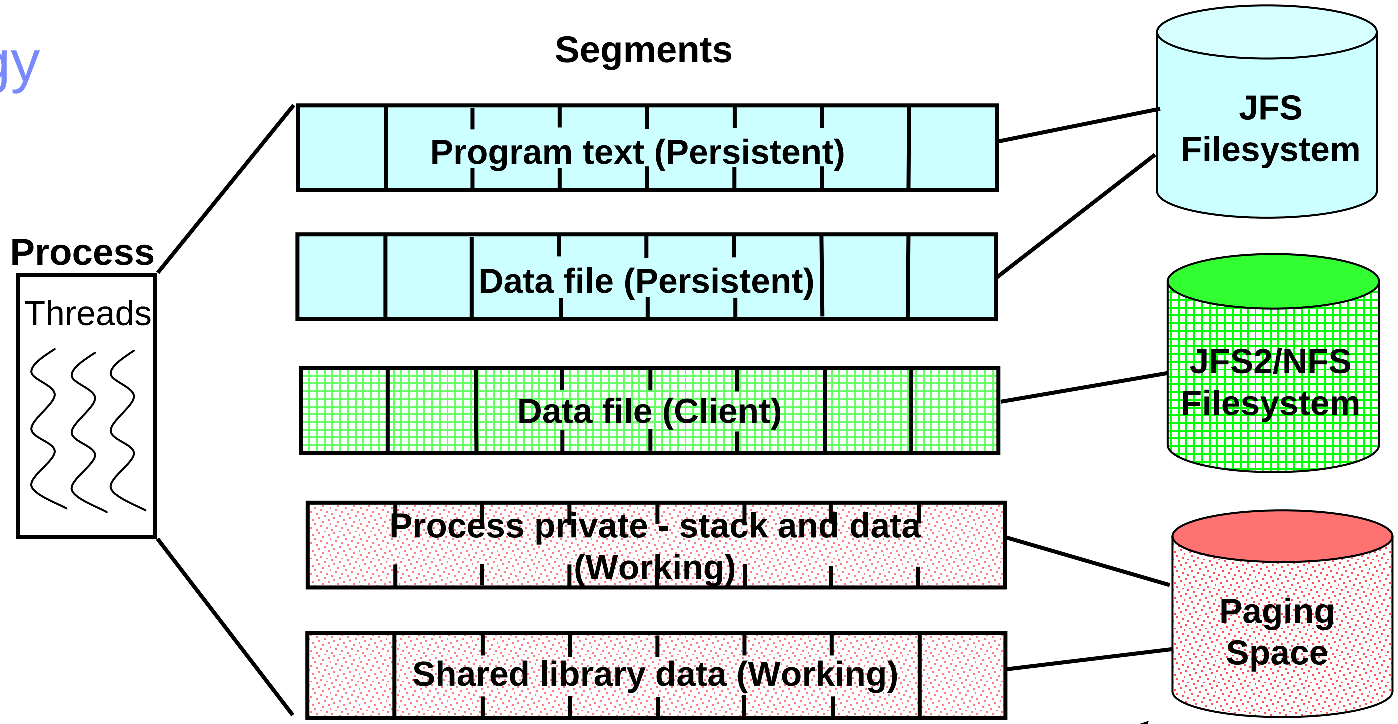
Address Spaces



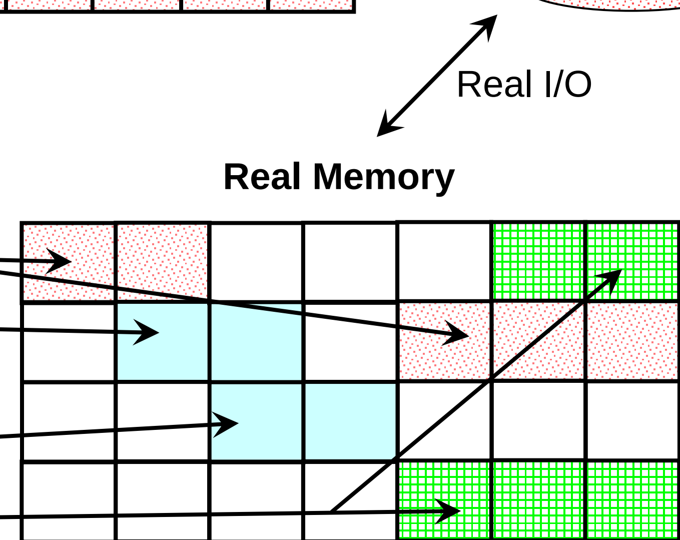
Process Address Space



VMM Terminology



- Segment Types:
  - Persistent
  - Client
  - Working



- Segment classification:
  - Computational:
    - Working segments
    - Program text
  - Non-computational (file memory):
    - Persistent segments
    - Client segments

Segment Identifiers

- Certain VSID values are reserved for special use
  - e.g. Kernel text segment (VSID = ESID = 0)

- Process ESID usage depends on:
  - Process type (32-bit or 64-bit)
  - Process state (user mode or kernel mode)
  - Kernel type (32-bit or 64-bit)

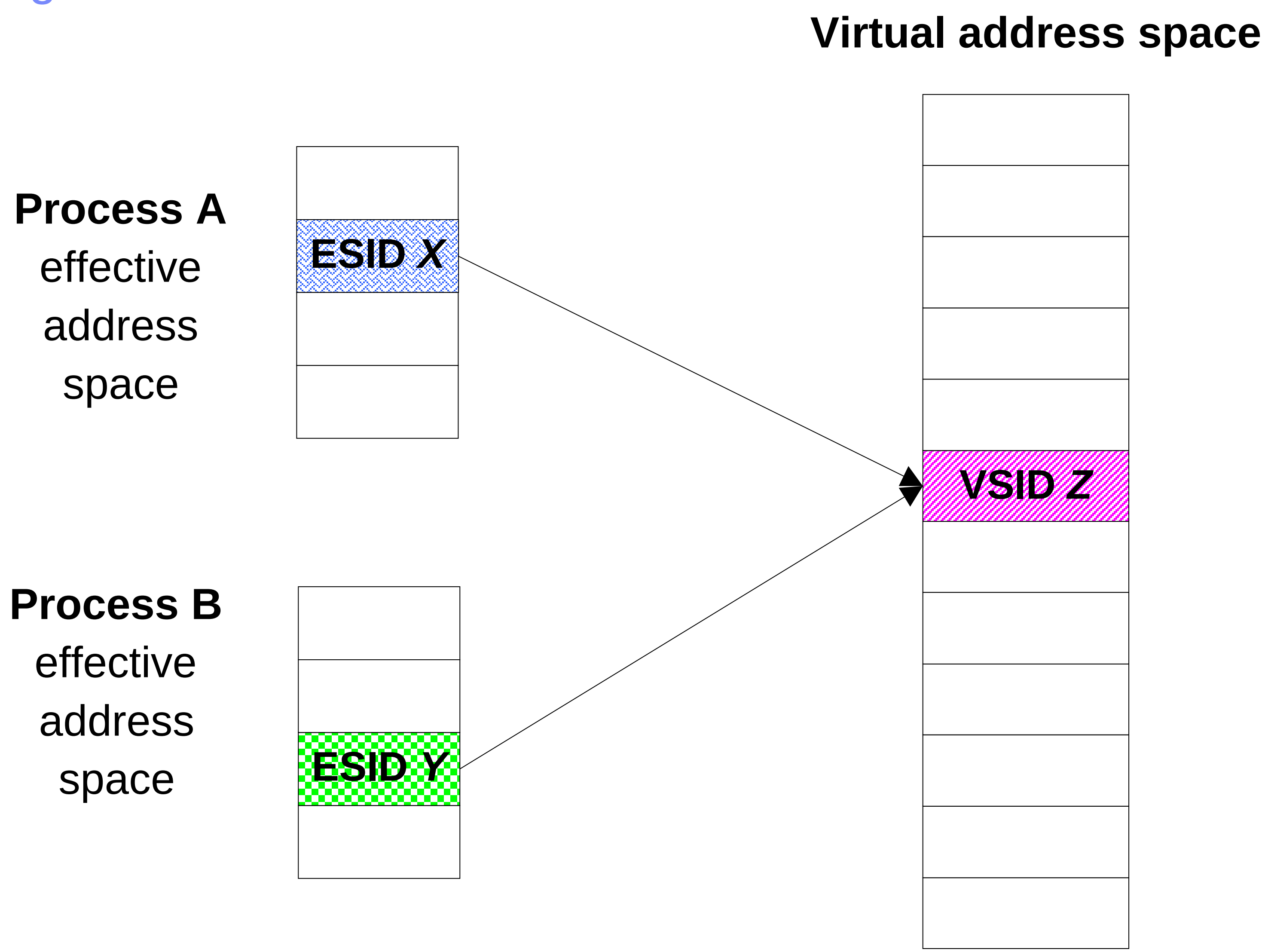
32-bit User Process Address Space

ESID (hexadecimal)	Description
0	Kernel segment
1	Program text
2	Process private (data, heap and stack)
3 - C	Shared data (**`shmat`** or **`mmap`**)
D	Shared library text
E	Shared data (**`shmat`** or **`mmap`**)
F	Shared library data

64-bit User Process Address Space

ESID (hexadecimal)	Description
0	Kernel segment
1	Reserved for system use
2	Process private (user mode loader data)
3 - C	Shared data (**shmat** or **mmap**)
D	Reserved (user mode loader)
E	Shared data (**shmat** or **mmap**)
F	Reserved (user mode loader)
10 – 6FFFFFFF	Application text, data, BSS and heap
70000000 – 7FFFFFFF	Default **shmat/mmap**
80000000 – 8FFFFFFF	Privately loaded modules
90000000 – 9FFFFFFF	Shared library text and data
F0000000 – FFFFFFFF	Application stack

Shared Segments



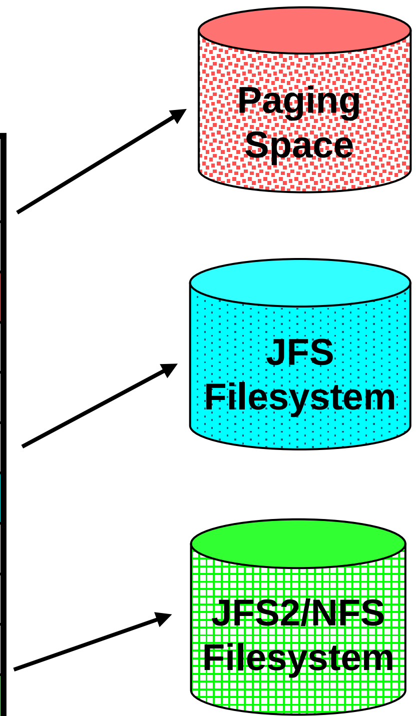
Managing Memory

- To manage memory, the VMM:
  - Manages the allocation of page frames
  - Resolves references to virtual memory pages that are not currently in RAM
- To accomplish these functions, the VMM:
  - Maintains a free list of available page frames
  - Uses a page replacement algorithm to determine which virtual memory pages, currently in RAM, will have their page frames reassigned to the free list
- The page replacement algorithm is called **`lrud`** (also referred to as the page stealer), which is a multi-threaded process
- Memory is divided into one or more memory pools. There is one **`lrud`** for each memory pool

Page Replacement Algorithms

Physical Address	Segment Type	Ref. Bit	Modified?
aaa1	W	On	Yes
aaa2	W	Off	Yes
aaa3	W	On	No
aaa4	W	Off	No
bbb1	P	On	Yes
bbb2	P	Off	Yes
bbb3	P	On	No
bbb4	P	Off	No
ccc1	C	On	Yes
ccc2	C	Off	Yes
ccc3	C	On	No
ccc4	C	Off	No

Initial PFT (excerpt)



Pages added to the free List

Physical Address
aaa2
aaa4
bbb2
bbb4
ccc2
ccc4

Resulting PFT (excerpt)

Physical Address	Segment Type	Ref. Bit	Modified?
aaa1	W		Yes
aaa3	W		No
bbb1	P		Yes
bbb3	P		No
ccc1	C		Yes
ccc3	C		No

Values for Persistent and Client Pages

- JFS pages are classified as persistent
  - The **numperm** value reflects number of non-computational pages in memory
- JFS2 and NFS pages are classified as client pages
  - The **numclient** value reflects number of client pages in memory
- Some command output:
  - Includes **numclient** in the **numperm** value (e.g., **vmstat**, **vmstat -v**)
  - Lists **numclient** and **numperm** values separately (e.g., **svmon**)

VMM Thresholds (1 of 2)

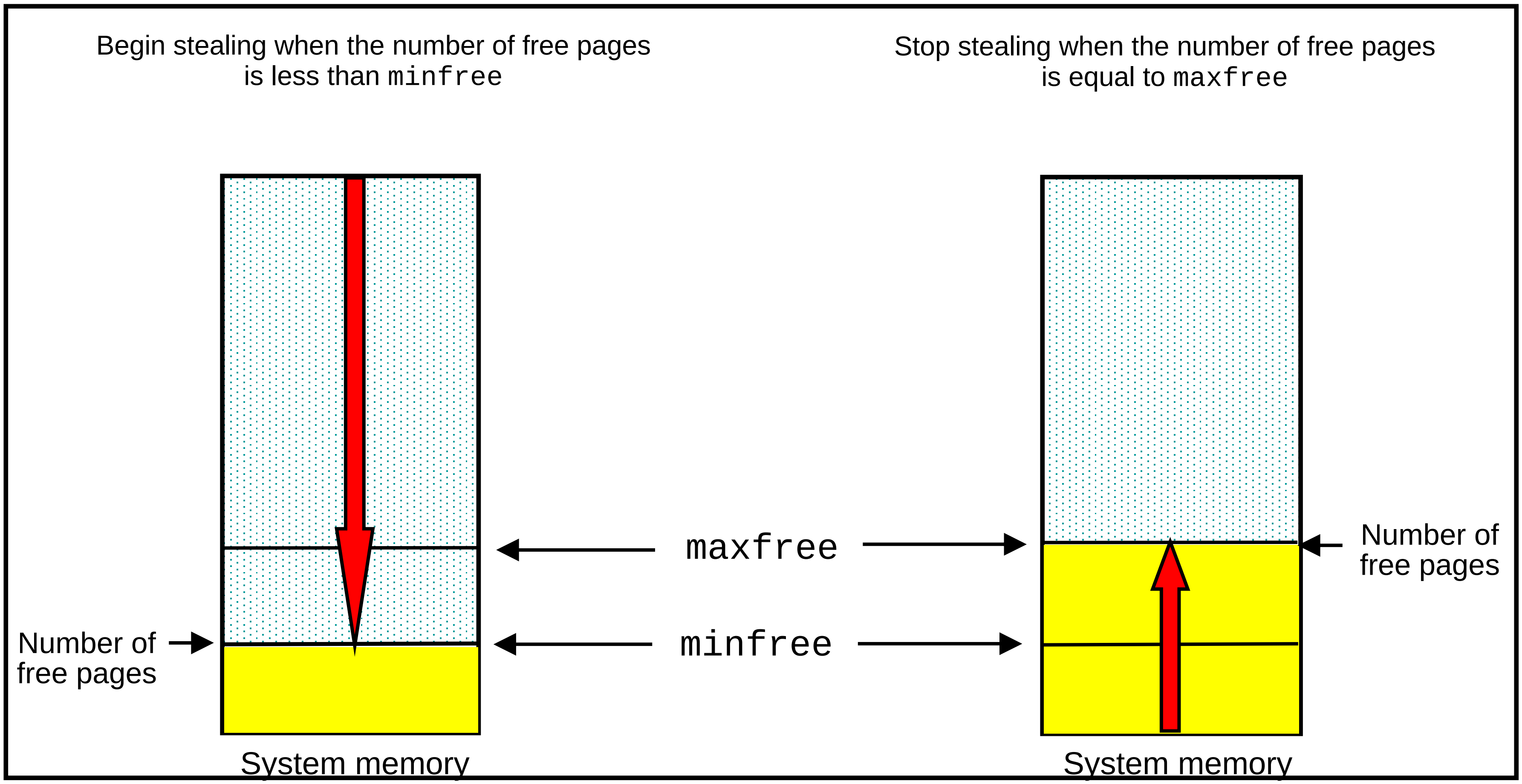
- The following **`vmo`** parameters ensure there are pages on the free list:
  - `minfree` - default 960 pages
  - `maxfree` - default 1088 pages
- The percentage of real memory that can be used by file pages (non-computational segments) is controlled by the following **`vmo`** parameters:
  - `minperm%`
    - AIX 5.2/5.3 - default 20%
    - AIX 6.1 - default 3%
  - `maxperm%`
    - AIX 5.2/5.3 - default 80%
    - AIX 6.1 - default 90%
  - `maxclient%`
    - AIX 5.2/5.3 - default 80%
    - AIX 6.1 - default 90%

VMM Thresholds (2 of 2)

- Other **`vmo`** parameters that affect page replacement are:
  - `strict_maxclient` (default 1)
  - `strict_maxperm` (default 0)
  - `lru_file_repage`
    - AIX 5.2/5.3 - default 1
    - AIX 6.1 - default 0

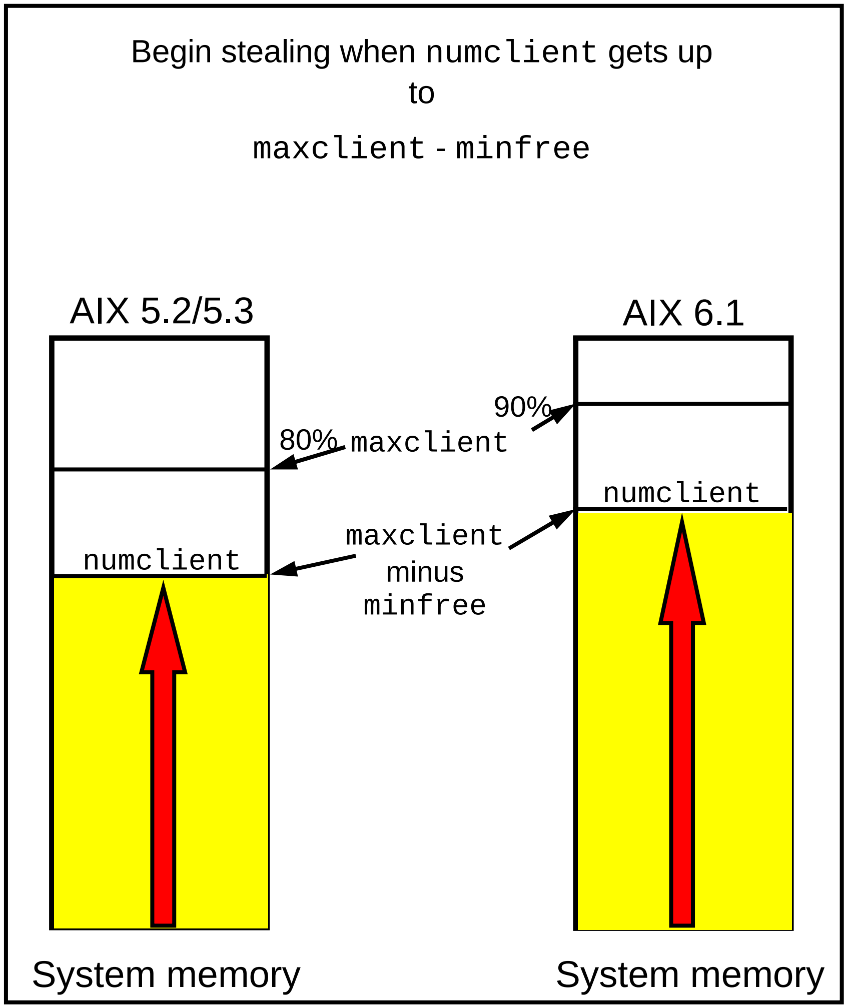
When to Steal Pages Based on Free Pages

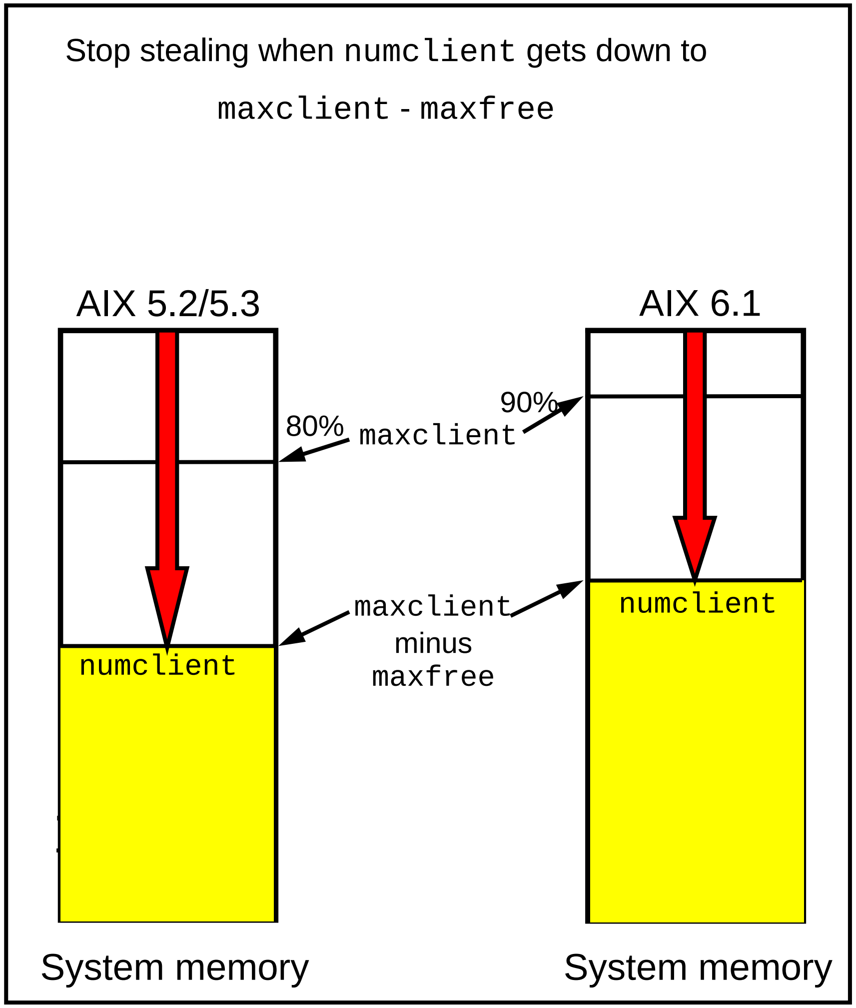
- The following **`vmo`** parameters ensure there are pages on the free list:
  - `minfree` (default 960 pages)
  - `maxfree` (default 1088 pages)



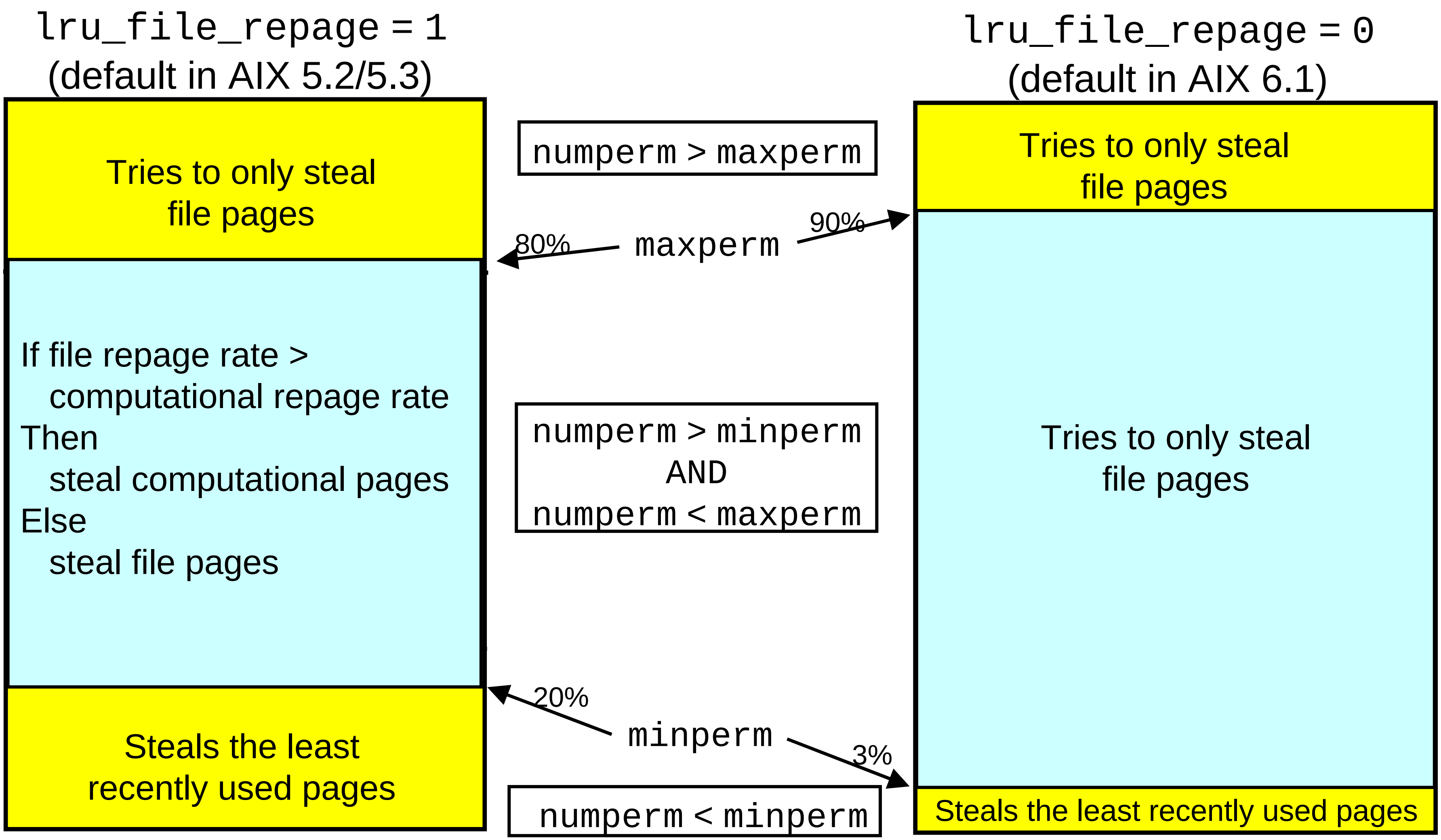
When to Steal Pages Based on Client Pages

- Assuming `strict_maxclient` = `1` (the default)
- Note: Only client pages are stolen in this situation





What Types of Pages are Stolen?



Note: File pages here mean BOTH client and persistent pages

How is Memory Being Used?

```
# vmstat -I 5

System configuration: lcpu=4 mem=1024MB

 kthr      memory                page                    faults              cpu
--------  -----------  --------------------------   ----------------- --------------
 r  b  p    avm    fre    fi   fo  pi  po   fr    sr   in     sy    cs    us  sy  id  wa
 1  0  0 187052  3219  11195   0   0   0  8720  9165  521 87646  7632    8  21  58  12
 1  0  0 187067  3214  4332    0   0   0  2697  2697  455 63884  4932    9  18  61  12
 1  0  0 187084  3144  3730    0   0   0  2374  2374  389 62610  5618   10  20  60  11
 2  0  0 187069  3006  4283    0   0   0  2634  2634  430 65111  6241   10  21  59  11
 1  0  0 187213  3048  5145    0   0   0  3979  75936 385 67500  4276    9  23  56  12
 0  1  0 187200  3140  14301   0   0   0  13494 13935 428 78735  3471    6  20  60  14
 1  0  0 187230  3188  13208   0   0   0  11605 11748 346 126253 12208   8  24  57  11
 1  0  0 187376  3135  3070    0   0   0  1092  1188  427 162036 29224  16  30  51   3
 1  0  0 187332  3618  4756    0   0   0  3390  3865  414 152360 21478  13  27  54   5
 1  0  0 187520  3244  4776    0   0   0  2351  2364  445 162840 30134  13  27  55   5
```

```
# svmon -G
               size       inuse        free         pin     virtual
memory       262144      259018        3126      108991      187230
pg space     131072        1876

               work        pers        clnt       other
pin           98745           0           0       10246
in use       187230           0       71788

PageSize   PoolSize       inuse        pgsp         pin     virtual
s    4 KB         -      147530        1876       23887       75742
m   64 KB         -        6968           0        5319        6968
```

Calculating Memory Usage (1 of 2)

- First `inuse` value includes memory used for 16 MB and 16 GB pages (even for unused pages)

```
# svmon -G
               size       inuse        free         pin     virtual
memory       262144      251966       10178      159298      160985
pg space     131072        2245

               work        pers        clnt       other
pin           83368           0           0       10394
in use       160985          21       29520

PageSize   PoolSize       inuse        pgsp         pin     virtual
s    4 KB         -       91102        2245       25634       61561
m   64 KB         -        5958           0        4258        5958
L   16 MB        16           1           0          16           1
```

- Actual total actively being used is obtained by adding `work`, `pers` and `clnt` values shown on the `in use` line

Calculating Memory Usage (2 of 2)

- When performing calculations, you need to convert to units of 4 KB

- Letters used by **`svmon`** to identify different page sizes
  - Do not associate the letter with a word
  - The letters used may change in future releases

Page size	svmon symbol	Number of 4 KB units
4 KB	s	1
64 KB	m	16
16 MB	L	4096
16 GB	S	4194304

Example Global Report (1 of 4)

```
# svmon -G
               size      inuse       free        pin    virtual
memory       262144     251966      10178     159298     160985
pg space     131072       2245

               work       pers       clnt      other
pin           83368          0          0      10394
in use       160985         21      29520

PageSize   PoolSize      inuse       pgsp        pin    virtual
s    4 KB         -      91102       2245      25634      61561
m   64 KB         -       5958          0       4258       5958
L   16 MB        16          1          0         16          1
```

- Total memory actively being used:
  160985 + 21 + 29520 = 190526
- Pinned 16 MB pages currently unused:
  (16 – 1) * 4096 = 61440
- Total physical memory currently allocated (i.e. not free):
  190526 + 61440 = 251966

Example Global Report (2 of 4)

```
# svmon -G
               size       inuse        free         pin     virtual
memory       262144      251966       10178      159298      160985
pg space     131072        2245

               work        pers        clnt       other
pin           83368           0           0       10394
in use       160985          21       29520

PageSize   PoolSize       inuse        pgsp         pin     virtual
s    4 KB         -       91102        2245       25634       61561
m   64 KB         -        5958           0        4258        5958
L   16 MB        16           1           0          16           1
```

- Total memory actively being used:
  160985 + 21 + 29520 = 190526
- Page size breakdown:
  91102 + (5958 * 16) + (1 * 4096) = 190526

Example Global Report (3 of 4)

```
# svmon -G
               size      inuse       free        pin     virtual
memory       262144     251966      10178     159298      160985
pg space     131072       2245

               work       pers       clnt      other
pin           83368          0          0      10394
in use       160985         21      29520

PageSize   PoolSize      inuse       pgsp        pin     virtual
s    4 KB         -      91102       2245      25634       61561
m   64 KB         -       5958          0       4258        5958
L   16 MB        16          1          0         16           1
```

Total virtual memory:

61561 + (5958 * 16) + (1 * 4096) = 160985

Example Global Report (4 of 4)

```
# svmon -G
               size      inuse       free        pin    virtual
memory       262144     251966      10178     159298     160985
pg space     131072       2245

               work       pers       clnt      other
pin           83368          0          0      10394
in use       160985         21      29520

PageSize   PoolSize      inuse       pgsp        pin    virtual
s    4 KB         -      91102       2245      25634      61561
m   64 KB         -       5958          0       4258       5958
L   16 MB        16          1          0         16          1
```

- First line **pin** value is memory that cannot be paged
  - Includes all 16 MB and 16 GB pages
- The in use **pin** value (83368 + 0 + 0 + 10394 in the example)
  - Excludes 16 MB and 16 GB pages on the 64-bit kernel
  - Includes used 16 MB pages on the 32-bit kernel

Is Memory Over Committed?

- **Memory is considered overcommitted if the number of pages currently in use exceeds the real memory pages available**
- **The number of pages currently in use is the sum of the:**
  - **Virtual pages**
  - **File cache pages**
- **If memory is over committed, then it is recommended to either:**
  - **Reduce the workload**
  - **Add more real memory**
- Example:

```
# svmon -G

                size       inuse        free         pin     virtual
memory        733184      731505        1679      191889      933823
pg space     1572864      282872

                work        pers        clnt
pin           191624           0         265
in use        689073           0       42432
```

```
  Virtual pages        = 933823 (3647 MB)
+ File cache pages     = 42432  (  166 MB)
--------------------------------------------------
  Total pages in use = 976255 (3813 MB)   vs.  Real memory = 733184 pages (2864 MB)
```

Pinned Memory Limits (1 of 3)

- The **`maxpin%`** tunable limits the amount of memory that can be pinned
  - Specified as a percentage of memory
  - Excludes memory used for non-pageable page sizes

- The **`vmo`** command reports both `maxpin%` and `maxpin` (the current value expressed in pages)

- Converted to an absolute value for each pageable page size
  - Stored as the number of pages that can't be pinned

- Kernel maintains a count of number of pages of each page size that are available for pinning

- A request to pin an address range will fail if the resulting number of pages available for pinning would drop below the reserved amount

Pinned Memory Limits (2 of 3)

- Pinned memory limits for a system can be viewed using **kdb**

```
(0)> pst *
PSX PSIZE NPAGES     PFAVAIL    PFRSVDBLKS NRSVD

00     4K 00015996  00011E0E  0000451E    00000000
01    64K 000017DD  0000073B  000004C5    00000000
02    16M 00000010  00000000  00000000    00000010
03    16G 00000000  00000000  00000000    00000000
```

- `PFRSVDBLKS` is the number of pages that cannot be pinned
  - Only applicable for page sizes that are pageable
  - Set based on `maxpin%` tunable
  - For example: `maxpin%` set to 80 (the default)
    0x15996 = 88470 x 20% = 17694 = 0x451E

- `PFAVAIL` is the number of pages that are available for pinning (i.e. the number of pages that are NOT pinned)

- 16 MB and 16 GB pages are not pageable

Pinned Memory Limits (3 of 3)

- Example:

```
# vmo -o maxpin
maxpin = 159378
```

- The `maxpin` value can be verified in **kdb**:
  - Total number of pages in system
    minus (total number of 16 MB pages * 4096)
    minus (total number of 16 GB pages * 4194304)
    minus PFRSVDBLKS values for 4 KB and 64 KB pages

```
(0)> pst *
PSX PSIZE NPAGES    PFAVAIL   PFRSVDBLKS NRSVD

00     4K 00015996  00011E0E  0000451E   00000000
01    64K 000017DD  0000073B  000004C5   00000000
02    16M 00000010  00000000  00000000   00000010
03    16G 00000000  00000000  00000000   00000000
```

  - Total number of good frames from **vmker** command in **kdb**

```
(0)> vmker | grep good
good page frames  (goodpages)  : 00040000
```

0x40000 – (0x10 * 4096) – 0 – 0x451E – (0x4C5 * 16) = 0x26E92 = 159378

Memory Values in `vmstat -v` (1 of 2)

```
# vmstat -v
             262144 memory pages
             172294 lruable pages
              10209 free pages
                  1 memory pools
              93746 pinned pages
               80.0 maxpin percentage
               . . . . . . . .
               . . . . . . . .
```

- Page counts shown in 4 KB units
- `memory pages` is amount of physical memory
- `lruable pages` is the number of pages that can be considered for page replacement
- `pinned pages` is the number of pinned pages
  - Should be similar to the in use pin count shown by **`svmon`**
  - Excludes 16 MB and 16 GB pages under the 64-bit kernel
  - Includes used 16 MB pages under the 32-bit kernel

Memory Values in `vmstat -v` (2 of 2)

```
# vmstat -v
                 . . . . . . . .
                 . . . . . . . .
                  3.0 minperm percentage
                 90.0 maxperm percentage
                 15.2 numperm percentage
                26298 file pages
                  0.0 compressed percentage
                    0 compressed pages
                 15.2 numclient percentage
                 90.0 maxclient percentage
                26277 client pages
```

- File cache limits (`minperm`, `maxperm`, and `maxclient`) are set as percentages of `lruable pages`
- **`vmstat –v`** shows limits and current usage (as both percentage and number of pages)

Significance of `lruable pages`

- The `lruable pages` value displayed by **`vmstat`** is the number of frames available for pageable memory
- This value will always be less than the total number of frames
- Does not include:
  - Memory allocated at boot time
  - VMM data structures
  - Pinned kernel text
  - Memory allocated for 16 MB and 16 GB pages
- The number of `lruable pages` available will change if:
  - The number of 16 MB or 16 GB pages is changed
  - A memory DLPAR add or remove is performed
- Remember that `minperm%`, `maxperm%`, and `maxclient%` are set as percentages of `lruable pages`

Examining Client Page Usage

- **svmon –G** reports total number of pages used for client segments
  - Includes computational segments (e.g. executables)

- Value reported by **vmstat –v** depends on AIX version

- In 64-bit kernel of AIX 5300-05 and above, **vmstat –v** reports number of non-computational client pages
  - i.e. pages used for file data rather than executable text
  - Value will likely be different than number of client pages reported by **svmon -G**

- In the 32-bit kernel of 5300-05, and both kernels of previous versions of AIX **vmstat –v** reports the same value as **svmon –G**
  - i.e. total number of client pages

Process Report

```
# svmon -P 278716

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
  278716 prog1            79959    71460        0    79955      N     N     Y

     PageSize                Inuse        Pin       Pgsp      Virtual
     s    4 KB                  23          4          0           19
     m   64 KB                 900        370          0          900
     L   16 MB                  16         16          0           16

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
    68b7         3 work working storage              L     16    16    0     16
    502d         d work shared library text          m    472     0    0    472
       0         0 work kernel segment               m    421   370    0    421
   11900         f work shared library data          m      7     0    0      7
    2373         2 work process private             sm     18     4    0     18
     8b1         1 clnt code,/dev/hd1:24617          s      3     0    -      -
   18929         e work shared memory segment       sm      1     0    0      1
   1c92d         - clnt /dev/hd3:666                 s      1     0    -      -
```

- Summary given in 4 KB units
- Page size distribution only shown if multiple sizes are in use
  - Single letter code is used as reference for each page size
- Segment usage information shown in page size units
  - Segments shown sorted by size, largest first
  - The `PSize` code letter(s) indicates size of pages in the segment

Process Memory Usage

```
# svmon -P 278716

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
  278716 prog1            79959    71460        0    79955      N     N     Y

     PageSize                Inuse        Pin       Pgsp      Virtual
     s    4 KB                  23          4          0           19
     m   64 KB                 900        370          0          900
     L   16 MB                  16         16          0           16

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
    68b7         3 work working storage              L     16    16    0    16
    502d         d work shared library text          m    472     0    0   472
       0         0 work kernel segment               m    421   370    0   421
   11900         f work shared library data          m      7     0    0     7
    2373         2 work process private             sm     18     4    0    18
     8b1         1 clnt code,/dev/hd1:24617          s      3     0    -     -
   18929         e work shared memory segment       sm      1     0    0     1
   1c92d         - clnt /dev/hd3:666                 s      1     0    -     -
```

- In use calculation:
  (16 * 4096) + (900 * 16) + 23 = 79959

- Pinned memory calculation:
  (16 * 4096) + (370 * 16) + 4 = 71460

Previous Style Process Report

```
# svmon -P 14044

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd LPage
   14044 5300-01comp      45555    41640        0    45553      N     N     Y

     PageSize        Inuse        Pin        Pgsp     Virtual
         4 KB         8691       4776           0        8689
        16 MB            9          9           0           9

    Vsid      Esid Type Description              LPage  Inuse   Pin Pgsp Virtual
   4c473         3 work shared memory segment (lgpg_  Y 36864 36864    0 36864
                    vsid=10410074)
       0         0 work kernel seg                   -   5958  4774    0  5958
    8022         d work shared library text          -   2709     0    0  2709
   702fc         2 work process private              -     13     2    0    13
   3c30f         f work shared library data          -      9     0    0     9
   7047c         1 clnt code,/dev/fslv00:49156       -      2     0    -     -
```

- Only 4 KB and 16 MB pages supported
  - LPage attribute indicates usage of 16 MB pages
- Segment usage statistics reported in 4 KB units
- For example:
  9 * (16 * 1024 * 1024) / 4096 = 36864

Previous Style Process Memory Usage

```
# svmon -P 14044

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  LPage
   14044 5300-01comp      45555    41640        0    45553      N     N      Y

     PageSize       Inuse        Pin       Pgsp    Virtual
         4 KB        8691       4776          0       8689
        16 MB           9          9          0          9

    Vsid      Esid Type Description              LPage  Inuse   Pin Pgsp Virtual
   4c473         3 work shared memory segment (lgpg_ Y  36864 36864    0 36864
                        vsid=10410074)
       0         0 work kernel seg                   -   5958  4774    0  5958
    8022         d work shared library text          -   2709     0    0  2709
   702fc         2 work process private              -     13     2    0    13
   3c30f         f work shared library data          -      9     0    0     9
   7047c         1 clnt code,/dev/fslv00:49156       -      2     0    -     -
```

- Calculation easier since segment information in 4 KB units
- In use calculation:
  36864 + 5958 + 2709 + 13 + 9 + 2 = 45555
- Pinned memory calculation:
  36864 + 4774 + 2 = 41640

Segment Information in `svmon`

- Vsid is the virtual segment ID
  - Each segment has a unique Vsid

- Esid is the effective segment ID
  - Shown in the process report
  - Only present when the segment is currently attached to the address space of the process
  - A '–' is used when the segment is not currently attached, or is not part of the user address space

- Type field indicates the nature of the segment
  - `clnt` = Client segment
  - `pers` = Persistent segment
  - `work` = Working segment
  - `mmap` = Memory map segment
  - `rmap` = Real mapping segment

Working Segments (1 of 2)

- Description field attempts to further classify the segment
  - Description based on Esid and Vsid information

- There are many different types of working segment

- Vsid 0 is the `kernel segment`
  - Contains the kernel text and some kernel data
  - Mapped by every process at Esid 0

- Descriptions of other kernel owned segments include:
  - `kernel heap`, `other kernel data`, `mbuf pool` and various VMM tables
  - The segments observed in **`svmon -S`** output will depend on kernel type (32-bit or 64-bit)

Working Segments (2 of 2)

- `process private`
  - For a 32-bit process contains data, BSS, heap and stack
  - For a 64-bit process contains loader data
- `application stack`
  - 64-bit process main stack area
- `text data BSS heap`
  - 64-bit process – the text segment will be of type `clnt` or `pers`, and list the device and inode number
  - Data, BSS and heap will be marked as type `work`
- `shared library text`, `shared library data`, `shared memory segment`
  - For both 32-bit and 64-bit processes
- `USLA heap`, `USLA text`
  - For 64-bit user processes

File Segments

- Segments of type `clnt` and `pers` are file segments
  – Description gives device name and inode number

- For example:

```
  Vsid    Esid Type Description          PSize  Inuse   Pin Pgsp Virtual
 1c92d       - clnt /dev/hd3:666             s      1     0    -       -
```

  – Device is **/dev/hd4**, inode number is 666

- Program text additionally identified as `code`

```
  Vsid    Esid Type Description          PSize  Inuse   Pin Pgsp Virtual
   8b1       1 clnt code,/dev/hd1:24617      s      3     0    -       -
```

- Can find the file using **`find`** or **`ncheck`**:
  Examples:

```
# find /home –inum 666 –xdev
# ncheck –i 666 /home
```

Segment Report

- One line of information about each segment on the system

- Description based on segment properties
  – May be blank

- Esid information not shown
  – Segment may not be in use by a process, or may be used by multiple processes using a different Esid in each

- Previous style report has `LPage` field instead of `PSize`

```
# svmon -S | more

     Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
    68b7         - work                              L     16    16    0     16
    7000         - work mbuf pool                    m   1704  1704    0   1704
    7001         - work kernel heap                  m   1328  1230    0   1328
    e3bf         - work                             sm  11482     0    0  11482
    6c00         - work kernel heap                  m    681   132    0    681
    e2df         - work mmap source                 sm   7966     0    0   7966
    502d         - work                              m    472     0    0    472
       0         - work kernel segment               m    421   370    0    421
   1c158         - work                              s   5632  5632    0   5632
    6800         - work kernel heap                  m    292   234    0    292
    8028         - work other kernel segments       sm   4096  4096    0   4096
    6b00         - work kernel heap                  m    211   203    0    211
    4000         - work page table area              s   2266    21 2245   2266
   120c3         - clnt /dev/hd2:82023               s   2031     0    -      -
. . . . . . . . .
```

Comparison of `svmon` Reports

- The **`-G`** report summarizes all segments on the system

- The **`-S`** report displays information on all segments
  – Each segment is listed once
  – Includes segments not associated with any process
  – By default, does not show which process is using a segment

- The **`-P`** report displays information on the segments being used by processes (or specific processes)
  – A segment may be listed multiple times (if used by multiple processes)
  – Cached files that are not currently open are not shown
  – Segments not associated with a process are not shown
  – Most kernel segments are not shown

Reports Included in PerfPMR Output

- The PerfPMR tool gathers multiple **`svmon`** reports
  - Reports generated by **`monitor.sh`**

- Four output files are created
  - Two at the start of data collection
  - Two at the end of data collection

- Global (**`svmon –G`**) and process (**`svmon –Pns`**) reports are stored in **svmon.before** and **svmon.after**

- Segment reports (**`svmon –lS`**) are stored in **svmon.before.S** and **svmon.after.S**
  - The **`–l`** flag displays Esid information (where possible) and process IDs of the processes using a segment
  - Not all segment types will have PID(s) listed

Shared Memory Segments and ipcs

- Shared memory segments can also be viewed using **ipcs**

- By default, all IPC information is displayed
  - Shared memory, message queues and semaphores

- Use the **–m** flag to only display shared memory information
  - Add the **–P** flag to see Vsid of each object

```
# ipcs -mP
IPC status from /dev/mem as of Wed Jan 20 23:42:14 CST 2010
T        ID     KEY        MODE       OWNER    GROUP
Shared Memory:
m   2097152 0xffffffff -----------     root   system
SID:0x18929  PINSIZE:0  LGPG: -
m   1048577 0xffffffff --rw-rw----     root   system
SID:0x10361  PINSIZE:0  LGPG: -
m   1048578 0x7800027a --rw-rw-rw-     root   system
SID:0x1d2ac  PINSIZE:0  LGPG: -
m         3 0xffffffff --rw-rw----     root   system
SID:0x103c1  PINSIZE:0  LGPG: -
```

Shared Memory Segments and PerfPMR

- Shared memory information is gathered by PerfPMR

- Output of **`ipcs –Sa`** included in the **config.sum** file
  - Provides more detail than **`ipcs -mP`**

```
# ipcs -Sa
IPC status from /dev/mem as of Wed Jan 20 23:43:09 CST 2010
T        ID     KEY        MODE        OWNER    GROUP  CREATOR   CGROUP NATTCH     SEGSZ  CPID   LPID    ATIME
  DTIME    CTIME
Shared Memory:
m   2097152 0xffffffff -----------     root   system     root   system      1      4096 278716 278716 23:15:58
  no-entry 23:15:58
SID :
0x18929
m   1048577 0xffffffff --rw-rw----     root   system     root   system      1     65536 295066 295066 23:41:13
  23:41:13 12:55:13
SID :
0x10361
m   1048578 0x7800027a --rw-rw-rw-     root   system     root   system      1  16777216 299162 299162 12:55:09
  no-entry 12:55:09
SID :
0x1d2ac
m         3 0xffffffff --rw-rw----     root   system     root   system      1     65536 295066 295066 23:41:13
  23:41:13 12:55:15
SID :
0x103c1
```

EXTSHM and Shared Memory Objects

- By default, each attach of a shared memory object consumes at least one segment of address space
  - 32-bit processes are limited to 11 concurrent attaches
  - Limit for 64-bit processes is considerably larger

- The EXTSHM facility increases the number of concurrently attached shared memory objects for 32-bit processes
  - Objects created with EXTSHM cannot be increased in size

- Use of EXTSHM creates additional segments
  - A working segment is created for each shared memory object
  - `mmap` segments are used to map the working segments into the process address space
  - This is reflected in **`svmon`** output

Without EXTSHM Example (1 of 2)

```
# svmon -P 14120

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
   14120 noextshm         10560     6267        0    10557      N     N     N

     PageSize      Inuse        Pin        Pgsp    Virtual
     s   4 KB      10560       6267           0      10557
     L  16 MB          0          0           0          0

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
       0         0 work kernel segment               s   7689  6265    0  7689
   6c05b         d work shared library text          s   2844     0    0  2844
   5c397         2 work process private              s     13     2    0    13
   6837a         f work shared library data          s      9     0    0     9
   2c3eb         1 clnt code,/dev/fslv03:20661       s      3     0    -     -
   58396         4 work shared memory segment        s      1     0    0     1
   4c393         3 work shared memory segment        s      1     0    0     1
```

Without EXTSHM Example (2 of 2)

```
# ipcs -Sa
IPC status from /dev/mem as of Wed Jan 31 20:42:19 CST 2007
T        ID     KEY        MODE       OWNER    GROUP  CREATOR   CGROUP NATTCH
  SEGSZ  CPID  LPID   ATIME    DTIME    CTIME
Shared Memory:
m    131072 0xffffffff --rw-rw----     root   system     root   system      1
   4096  9402  9402 20:41:29 20:41:29 17:15:31
SID :
0x782fe
m         3 0xffffffff --rw-rw----     root   system     root   system      1
   4096  9402  9402 20:41:29 20:41:29 17:15:32
SID :
0x48312
m         4 0x0d0009e2 --rw-rw----     root   system     root   system      1
   1440 11248 10650 19:59:42 19:59:43 17:19:11
SID :
0x24349
m         5 0xffffffff -----------     root   system     root   system      1
   4096 14120 14120 20:40:21 no-entry 20:40:21
SID :
0x4c393
m         6 0xffffffff -----------     root   system     root   system      1
   4096 14120 14120 20:40:21 no-entry 20:40:21
SID :
0x58396
```

With EXTSHM Example (1 of 2)

```
# svmon -P 14130

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
   14130 extshm           10559     6267        0    10556      N     N     N

     PageSize       Inuse         Pin        Pgsp     Virtual
     s   4 KB       10559        6267           0       10556
     L  16 MB           0           0           0           0

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
       0         0 work kernel segment               s   7689  6265    0  7689
   6c05b         d work shared library text          s   2844     0    0  2844
   5c397         2 work process private              s     12     2    0    12
   58396         f work shared library data          s      9     0    0     9
   4c393         1 clnt code,/dev/fslv03:20659       s      3     0    -     -
   243e9         - work mmap source                  s      1     0    0     1
   6837a         - work mmap source                  s      1     0    0     1
   1c3e7         3 mmap maps 2 source(s)             s      0     0    -     -
```

With EXTSHM Example (2 of 2)

```
# ipcs -Sa
IPC status from /dev/mem as of Wed Jan 31 20:46:53 CST 2007
T        ID      KEY        MODE       OWNER    GROUP  CREATOR   CGROUP NATTCH
  SEGSZ  CPID  LPID   ATIME    DTIME    CTIME
Shared Memory:
m    131072 0xffffffff --rw-rw----     root   system     root   system      1
   4096  9402  9402 20:45:29 20:45:29 17:15:31
SID :
0x782fe
m         3 0xffffffff --rw-rw----     root   system     root   system      1
   4096  9402  9402 20:45:29 20:45:29 17:15:32
SID :
0x48312
m         4 0x0d0009e2 --rw-rw----     root   system     root   system      1
   1440 11248 10650 19:59:42 19:59:43 17:19:11
SID :
0x24349
m    131077 0xffffffff -----------     root   system     root   system      1
   4096 14130 14130 20:45:12 no-entry 20:45:12
SID :
0x243e9
m    131078 0xffffffff -----------     root   system     root   system      1
   4096 14130 14130 20:45:12 no-entry 20:45:12
SID :
0x6837a
```

Description Based on Usage

```
# svmon -P 278752

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
  278752 duplo2           14812     8089        0    14809      N     N     N

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
       0         0 work kernel segment               s  11888  8086    0 11888
   7709d         d work shared library text          s   2892     0    0  2892
   7835a         2 work process private              s     15     3    0    15
   4c3b7         f work shared library data          s     12     0    0    12
   743b9         1 clnt code,/dev/fslv03:20656       s      3     0    -     -
   243ad         - work mmap source                  s      1     0    0     1
   783ba         - work mmap source                  s      1     0    0     1
   583b2         3 mmap maps 2 source(s)             s      0     0    -     -
# svmon -P 295132

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
  295132 duplo            14815     8089        0    14812      N     N     N

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
       0         0 work kernel segment               s  11891  8086    0 11891
   7709d         d work shared library text          s   2892     0    0  2892
    4365         2 work process private              s     15     3    0    15
   7c3bb         f work shared library data          s     12     0    0    12
   5c3b3         1 clnt code,/dev/fslv03:20657       s      3     0    -     -
   783ba         3 work shared memory segment        s      1     0    0     1
   243ad         4 work shared memory segment        s      1     0    0     1
```

Identifying Shared Segments

- The **-l** flag can be used in the process report to list the PID(s) of all the processes sharing a segment
  - Not shown for shared library text segments or segment 0

```
# svmon -P 18616 -l

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
   18616 duplo2           10558     6271        0    10555      N     N     N

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
       0         0 work kernel segment               s   7689  6269    0  7689
                    System segment
   6c05b         d work shared library text          s   2840     0    0  2840
                    Shared library text segment
    c3e3         f work shared library data          s     12     0    0    12
                    pid(s)=18616
   3c3cf         2 work process private              s     12     2    0    12
                    pid(s)=18616
    4361         1 clnt code,/dev/fslv03:20656       s      3     0    -     -
                    pid(s)=18616
   3036c         3 work mmap source                  s      1     0    0     1
                    pid(s)=18616, 15556
   7435d         4 work mmap source                  s      1     0    0     1
                    pid(s)=18616, 15556
```

Large Shared Memory Regions (1 of 2)

```
# svmon -P 167948

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
  167948 a.out           211364     8090        0   211362      N     N     N

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
   1c323         5 work unused segment               s  65536     0    0 65536
   14321         4 work unused segment               s  65536     0    0 65536
   6031c         3 work shared memory segment        s  65536     0    0 65536
       0         0 work kernel segment               s  11884  8087    0 11884
   7709d         d work shared library text          s   2844     0    0  2844
   7831a         2 work process private              s     17     3    0    17
   4c317         f work shared library data          s      9     0    0     9
   7c37b         1 clnt code,/dev/fslv03:20646       s      2     0    -     -
```

```
# ipcs -Sa
IPC status from /dev/mem as of Thu Jan 11 16:53:45 CST 2007
T        ID     KEY        MODE       OWNER    GROUP  CREATOR   CGROUP NATTCH
  SEGSZ  CPID  LPID   ATIME    DTIME    CTIME
Shared Memory:
m   1048576 0xffffffff --rw-rw----     root    system     root    system      1
   4096 213132 213132 16:52:36 16:52:36 16:20:36
SID :
0x482b6
m         5 0xffffffff -----------     root    system     root    system      1 80
5306368 167948 167948 16:51:50 no-entry 16:51:50
SID :
0x6031c 0x14321 0x1c323
```

Large Shared Memory Regions (2 of 2)

```
# svmon -P 262308

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
  262308 a.out64         210249     8109        0   210236      Y     N     N

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
   18382  70000001 work default shmat/mmap           s  65536     0    0 65536
    8386  70000002 work default shmat/mmap           s  65536     0    0 65536
     384  70000000 work default shmat/mmap           s  65536     0    0 65536
       0         0 work kernel segment               s  11900  8087    0 11900
    4001  9ffffffd work shared library               s   1252     0    0  1252
    30a0  90000000 work shared library text          s    376     0    0   376
   2414d  90020014 work shared library               s     39     0    0    39
   1c323 f00000002 work process private              s     28    22    0    28
   6031c  9001000a work shared library data          s     13     0    0    13
   34149  9fffffff clnt USLA text,/dev/hd2:3031      s     10     0    -     -
   58016  9ffffffe work shared library               s      9     0    0     9
   4c317  8fffffff work private load data            s      4     0    0     4
    c387  80020014 work USLA heap                    s      4     0    0     4
   7c37b        10 clnt text data BSS heap,          s      3     0    -     -
                        /dev/fslv03:20646
   7831a  ffffffff work application stack            s      2     0    0     2
   40374        11 work text data BSS heap           s      1     0    0     1
```

Two Instances of a Program

```
# svmon -P 266254

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
  266254 ksh              15032     8090        0    14971      N     N     N

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
       0         0 work kernel segment               s  11901  8087    0 11901
   7709d         d work shared library text          s   2936     0    0  2936
    c387         2 work process private              s    110     3    0   110
   600bc         1 clnt code,/dev/hd2:3108           s     56     0    -     -
   18382         f work shared library data          s     24     0    0    24
   48316         - clnt /dev/hd4:1709                s      3     0    -     -
   3802a         - clnt /dev/hd2:13035               s      2     0    -     -

# svmon -P 291004

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
  291004 ksh              15016     8090        0    14957      N     N     N

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
       0         0 work kernel segment               s  11898  8087    0 11898
   7709d         d work shared library text          s   2936     0    0  2936
   40374         2 work process private              s    100     3    0   100
   600bc         1 clnt code,/dev/hd2:3108           s     56     0    -     -
   6031c         f work shared library data          s     23     0    0    23
   3802a         - clnt /dev/hd2:13035               s      2     0    -     -
    4385         - clnt /dev/hd1:52                  s      1     0    -     -
```

Two Processes Accessing the Same File

```
# svmon -P 11376

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
   11376 more             10600     6271        0    10583      N     N     N

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
       0         0 work kernel segment               s   7688  6269    0  7688
  6c05b          d work text or shared-lib code seg  s   2830     0    0  2830
  103c4          f work working storage              s     32     0    0    32
  7c3bf          3 work working storage              s     19     0    0    19
  683ba          1 clnt code,/dev/hd2:3006           s     16     0    -     -
   83c2          2 work process private              s     14     2    0    14
  48372          - clnt /dev/fslv03:20658            s      1     0    -     -

# svmon -P 17036

-------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
   17036 pg               10578     6271        0    10569      N     N     N

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
       0         0 work kernel segment               s   7688  6269    0  7688
  6c05b          d work text or shared-lib code seg  s   2830     0    0  2830
  183c6          f work working storage              s     31     0    0    31
   c3c3          2 work process private              s     13     2    0    13
  1c3c7          1 clnt code,/dev/hd2:165            s      8     0    -     -
  643b9          3 work working storage              s      7     0    0     7
  48372          - clnt /dev/fslv03:20658            s      1     0    -     -
```

Determining Memory Usage

- The Process report sorts based on memory usage
  - Largest memory user displayed first
- Remember that a process is counted as using all pages from all segments it references
  - Even if the segments are shared by other processes
  - Even if the process has only really used a small number of pages from the segment

Determining Paging Space Usage (1 of 2)

- The process report can be changed with the **-g** flag to sort based on paging space usage
  - This is not included in PerfPMR output

```
# svmon -Pg

     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
  274572 rpc.statd        14103     5927      582    14650      N     Y     N

     PageSize                Inuse        Pin       Pgsp    Virtual
     s    4 KB                   7          7        326        362
     m   64 KB                 881        370         16        893
     L   16 MB                   0          0          0          0

    Vsid      Esid Type Description              PSize  Inuse   Pin Pgsp Virtual
       0         0 work kernel segment              m    410   370   15   421
   1a24b         f work shared library data        sm      0     0  186   213
    c23d         3 work working storage            sm      0     0  108   110
   11240         2 work process private            sm      4     4   20    24
    502d         d work shared library text         m    471     0    1   472
   112a0         - work                             s      3     3   12    15
   19248         5 work working storage            sm      0     0    0     0
   1c24d         4 work working storage            sm      0     0    0     0
    e23f         1 clnt code,/dev/hd2:66191         s      0     0    -     -

 -------------------------------------------------------------------------------
     Pid Command          Inuse      Pin     Pgsp  Virtual 64-bit Mthrd  16MB
  221296 sendmail         14183     5924      489    14562      N     N     N
```

Determining Paging Space Usage (2 of 2)

- Can determine paging space usage from **svmon** reports by use of **grep**, **egrep**, **awk**, and **sort**
- For process report, suggest using:

```
grep –p Command svmon.before | egrep –v '(^-|16MB|^$)' | \
awk '{print $5,$1,$2}' | sort –nr
```

- Output is sorted based on paging space usage
  - Each line displays paging space usage, process ID and command name

```
1311 12486 xmwlm
1297 14708 rpc.statd
1294 9184 nfsrgyd
1257 13936 rpc.mountd
. . . . .
```

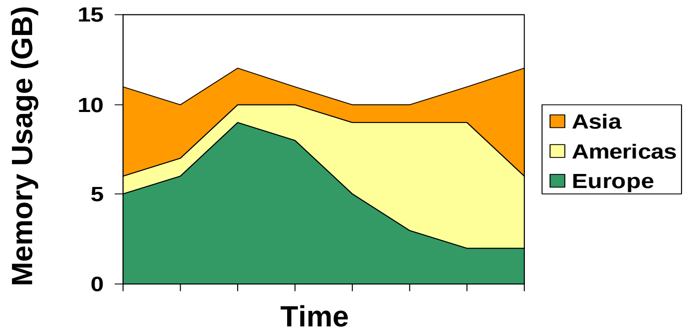
- Segment report is much more complicated
  - But paging space normally investigated on a per-process basis anyway

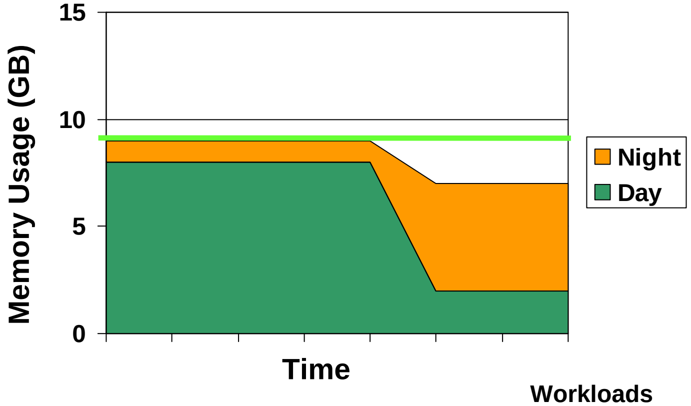
Active Memory Sharing

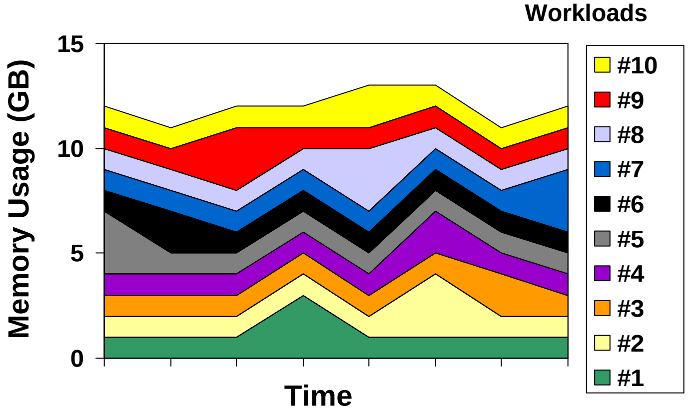
- Active Memory Sharing (AMS) is a new PowerVM virtualization technology added in May 2009
- Allows a group of partitions to share a single pool of physical memory
- The hypervisor dynamically allocates physical memory from the pool to the partitions, based on demand
  – Allocation is at the page level of granularity, not the system LMB size
- Can improve overall utilization of physical memory resources
- Allows over-commitment of logical memory
  – Overflow stored on VIOS-managed paging devices

Why Use Active Memory Sharing?

- AMS dynamically optimizes memory over multiple LPARs based on workload:
  - – Different workload peaks due to timezones
  - – Mixed workloads peak at different times of day
  - – Ideal for consolidated workloads with low or sporadic memory requirements
- No user intervention required after initial configuration







Monitoring AMS in AIX (1 of 2)

- The **`vmstat`** command has been updated to display hypervisor paging information

```
# vmstat -h 2 3

System configuration: lcpu=2 mem=4096MB ent=0.50 mmode=shared mpsz=6.00GB

kthr    memory              page              faults              cpu             hypv-page
----- ----------- ------------------------ ------------ ----------------------- -----------------------
 r  b   avm   fre  re  pi  po  fr   sr  cy  in   sy  cs us sy id wa    pc    ec  hpi  hpit   pmem   loan
 0  0 175395 837661   0   0   0   0    0   0   6   53 162  0  0 99  0  0.00   0.8     1     3   3.58   0.42
 0  0 175395 834589   0   0   0   0    0   0   2   11 156  0  1 99  0  0.01   1.5     0     0   3.57   0.43
 0  0 175395 834566   0   0   0   0    0   0   1   10 154  0  0 99  0  0.00   0.7     0     0   3.57   0.43
```

  – `mmode` = partition memory mode, shared or dedicated
  – `mpsz` = memory pool size
  – `hpi` = number of hypervisor page ins
  – `hpit` = time spent waiting for hypervisor page ins (in milliseconds)
  – `pmem` = physical memory backing the partition
  – `loan` = amount of logical memory loaned to the hypervisor

- If `pmem` + `loan` is less than partition logical memory, then pages have been stolen by the hypervisor

Monitoring AMS in AIX (2 of 2)

- Adding the **–h** flag to usage of **vmstat -v** shows four additional lines

```
# vmstat -v -h
            1048576 memory pages
            1002276 lruable pages
             801166 free pages
                  1 memory pools
             127821 pinned pages
               80.0 maxpin percentage
                3.0 minperm percentage
               90.0 maxperm percentage
                0.1 numperm percentage
               2000 file pages
                0.0 compressed percentage
                  0 compressed pages
                0.1 numclient percentage
               90.0 maxclient percentage
               2000 client pages
                  0 remote pageouts scheduled
                  0 pending disk I/Os blocked with no pbuf
              54736 paging space I/Os blocked with no psbuf
               2484 filesystem I/Os blocked with no fsbuf
                  0 client filesystem I/Os blocked with no fsbuf
               1209 external pager filesystem I/Os blocked with no fsbuf
             206792 Virtualized Partition Memory Page Faults
             777186 Time resolving virtualized partition memory page faults
             146765 Number of 4k page frames loaned
                 13 Percentage of partition memory loaned
```

Session Summary

- Define virtual memory concepts and terminology and explain their impact on memory based performance issues
- Calculate and categorize the memory in use on the system
- Identify which processes are using the most memory
- Determine if a system has enough memory

Gracias from the Texan!