

Taming Your Storage Hungry Linuxen Using CMM(A)

David Kreuter
Denver SHARE
Session 9272

Taming Storage Hungry Linux Virtual Machines

- Problem case study
- CMM 1 – software controls service machine communication
- CMMA (CMM – 2) – hardware storage key bit settings
- Descriptions
- Scenarios
- Performance data

Tools

- CP commands
 - INDICATE, QUERY, XAUTOLOG, FORCE, SET
- CMS: REXX, PIPELINES
- CP MONITOR DATA
- Velocity products
- Linux commands:
 - cat, lsmmod, ls, grep, vi, top, nice, cp, mv, rm

CMM1

- Linux on System z support for CMM1 is available in:
 - Novell SUSE Linux Enterprise Server 9 (SLES9) SP3 since kernel level: kernel-s390(x)-2.6.5-7.257 dated 2006-05-16
 - Novell SUSE Linux Enterprise Server 10 (SLES10) since GA
 - Red Hat RHEL4 U7 2.6.9-73 (includes Out of Memory Notifier)
 - Red Hat RHEL5.1 2.6.18-53 (includes Out of Memory Notifier)
- In z/VM: 5.3.0 and beyond
 - In 5.2.0 CMS APAR is required, VM64085, for full functionality

CMM1

- The VM Resource Manager service machine
- Linux drivers for CMM processing and message handling.
- Used effectively can reduce Linux storage footprint

CMM Linux Mechanics

- Load the CMM module with modprobe or insmod
 - *Not compiled in kernel is Novell SLES10*
- Parameter passing
- Checking the parameters after loading
- Dynamic and static loading methods

Dynamically loading the CMM module in Linux

```
# lsmod | grep cmm
```

```
# modprobe cmm
```

```
#lsmod
```

| Module | Size | Used by |
|----------|-------|---------|
| cmm | 33024 | 0 |
| smsgiucv | 24080 | 1 cmm |
| iucv | 47704 | 1 |

```
:
```

Dynamically loading the CMM module in Linux specifying the sender

```
# modprobe cmm sender=SOMEVM
```

```
#
```

```
# lsmod
```

| Module | Size | Used by |
|----------|-------|---------|
| cmm | 33024 | 0 |
| smsgiucv | 24080 | 1 cmm |
| iucv | 47704 | 1 |
| : | | |

Dynamically checking the parameters

```
# cat /sys/module/cmm/parameters/sender  
SOMEVM  
#
```

Kernel level and distribution

```
# cat /proc/version /etc/SuSE*  
Linux version 2.6.16.60-0.21-default (geeko@buildhost)  
(gcc version 4.1.2 20070115 (SUSE Linux)) #1  
SMP Tue May 6 12:41:02 UTC 2008  
  
SUSE Linux Enterprise Server 10 (s390x)  
VERSION = 10  
PATCHLEVEL = 2  
  
#
```

Not compiled in the kernel

```
grep -i cmm /boot/config-*  
CONFIG_CMM=m  
CONFIG_CMM_PROC=y  
CONFIG_CMM_IUCV=y
```

In SLES SP10 cmm is not
compiled into the kernel

Automatically loading the cmm module

- Train the kernel in `/etc/sysconfig/kernel`
- Pass parameters in `/etc/modprobe.conf.local`

```
~ grep -i cmm /etc/sysconfig/kernel
MODULES_LOADED_ON_BOOT="vmcp cmm"

~ cat /etc/modprobe.conf.local
#
# please add local extensions to this file
#
options cmm sender=OTHERVM
```

Checking after boot time

```
~ cat /sys/module/cmm/parameters/sender  
OTHERVM
```

The VM Resource Manager

- Workload manager for z/VM
- Can be used to dynamically adjust virtual machine CPU consumption and I/O usage
- Used to message Linux virtual machines when using CMM
- Runs in the VMRMSVM service machine as supplied by IBM.
- One configuration file.
- Logs to a file.

The VM RMSVM Under the Hood

- Use CP MONITOR SAMPE data to determine:
 - Memory constraints
 - How much memory to instruct its' Linux partner to release
- “Kicks in”
- Requires careful monitoring – can have profound positive impact but can also hurt

The VM Resource Manager: startup

```
xautolog vmrmsvm
```

```
Command accepted
```

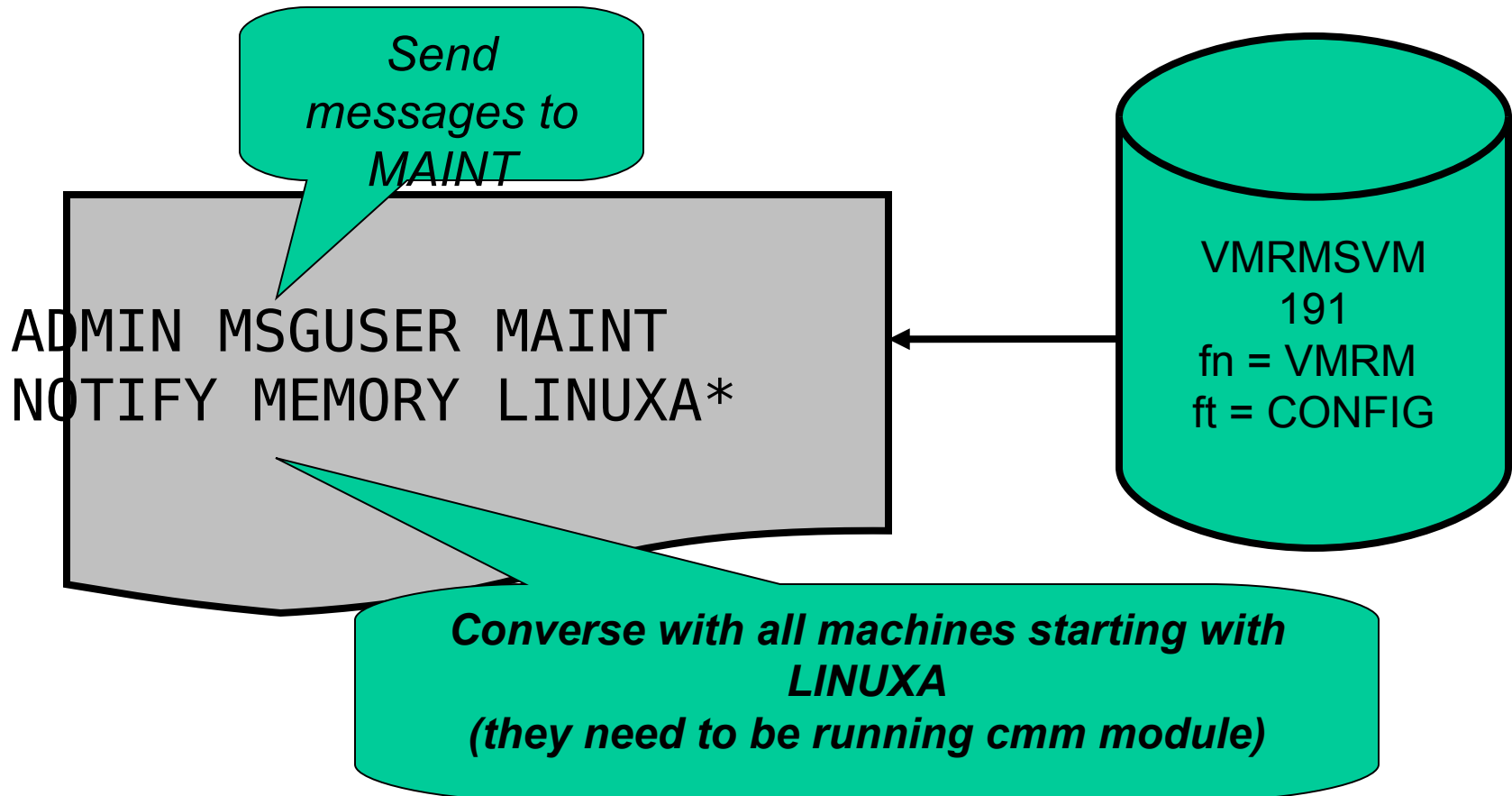
```
AUTO LOGON   ***           VMRMSVM  USERS = 62
```

```
HCPCLS6056I XAUTOLOG information for VMRMSVM: The IPL command is  
verified by the IPL command processor.
```

```
12:53:38 * MSG FROM VMRMSVM : IRMSER0023I VM Resource Manager Service  
Virtual
```

```
Machine initialization complete. Proceeding to connect to Monitor.
```


Configuration file: NOTIFY statement in the VMRM CONFIG



The VM Resource Manager: CMM notifications

1

```
query mon sample

MONITOR SAMPLE ACTIVE

          INTERVAL      1
MINUTES

          RATE         1.00
SECONDS

MONITOR DCSS NAME - MONDCSS

CONFIGURATION SIZE      1000
LIMIT                1 MINUTES
CONFIGURATION AREA IS FREE

USERS CONNECTED TO *MONITOR -
ESAWRITE VMPMSVM
```

2

```

:
MONITOR  DOMAIN ENABLED

SYSTEM  DOMAIN ENABLED

PROCESSOR DOMAIN ENABLED

STORAGE  DOMAIN ENABLED

USER     DOMAIN ENABLED

        ALL USERS ENABLED

I/O     DOMAIN ENABLED

        ALL DEVICES ENABLED
```

RE 92

The VM Resource Manager: orderly termination

*CMS immediate
command*

```
cp send vmrmsvm hmonitor
```

```
12:54:28 * MSG FROM VM RMSVM : IRMMON0026I VM Resource Manager  
processing of monitor records ended. Pipe RC= 0
```

```
12:54:28 * MSG FROM VM RMSVM : IRMSER0012I VM Resource Manager Service  
Virtual Machine shutdown in progress
```

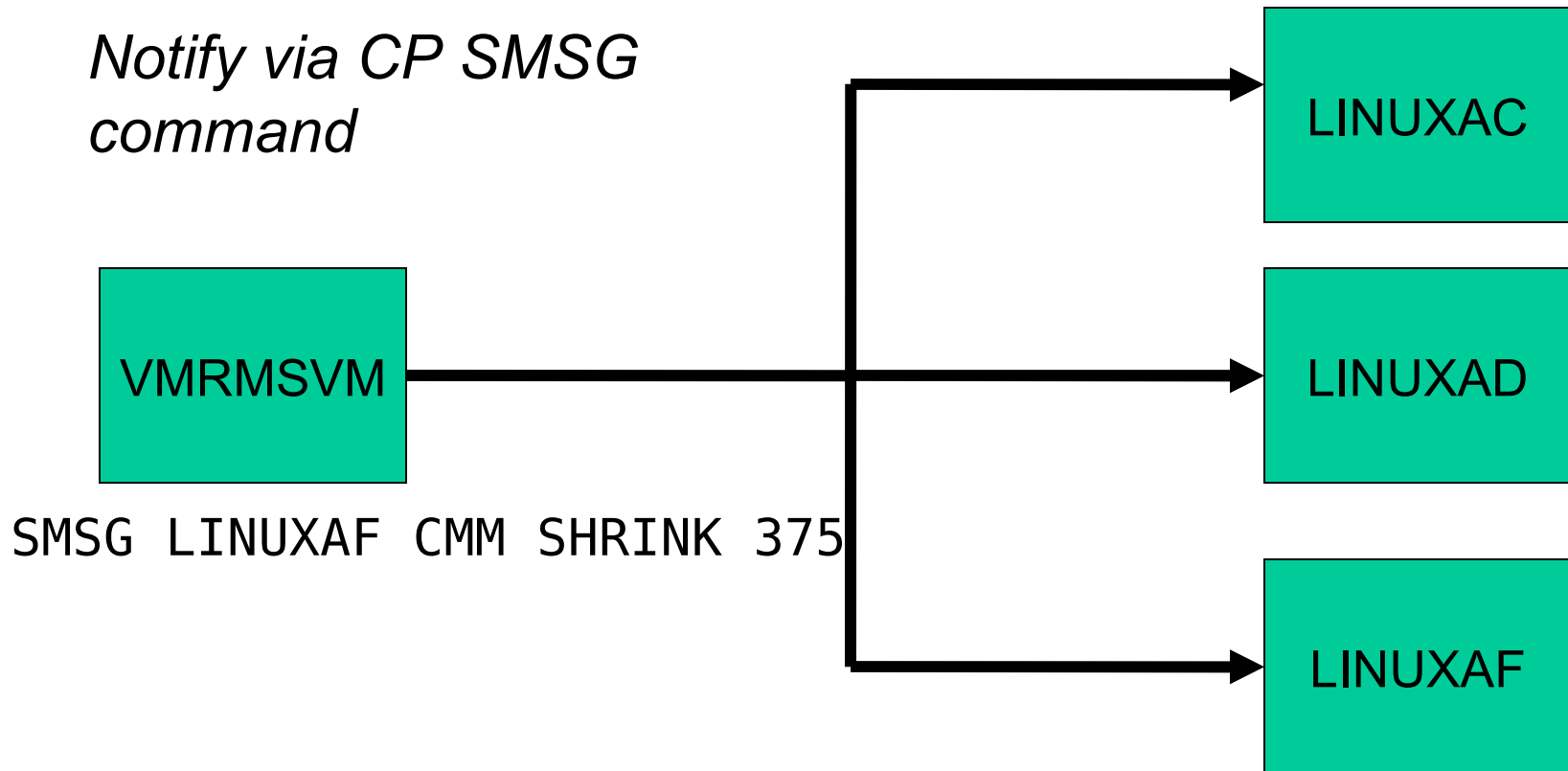
```
12:54:28 * MSG FROM VM RMSVM : IRMSER0027I VM Resource Manager Service  
Virtual Machine shutdown complete
```

```
VM RMSVM : 12:54:28 0 RC FROM IRMSERV
```

```
VM RMSVM : 12:54:28 Ready: T=0 17/0 18 12:54:28
```

VMRMSVM and Linux interaction

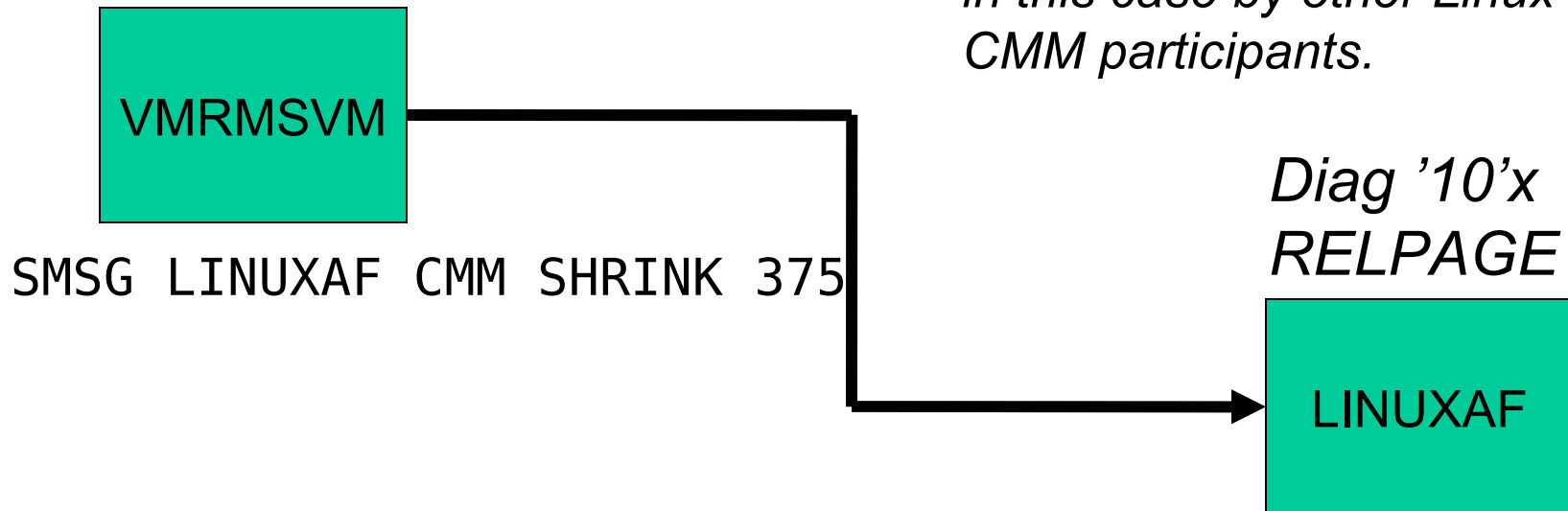
Notify via CP SMSG command



VMRMSVM and Linux interaction

Notify via CP SMSG command

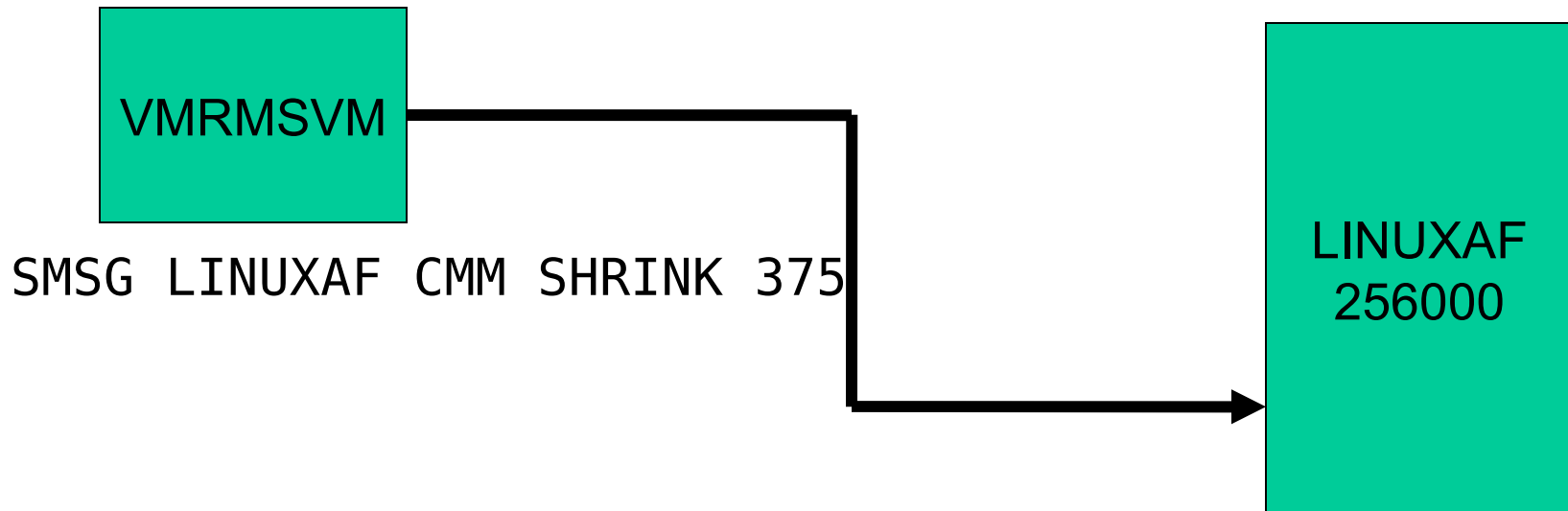
Relpage means that the virtual machine doesn't need the page backed up by CP. Therefore it can be reused; in this case by other Linux CMM participants.



Page SHRINK 1

*Notify via CP SMSG
command*

*Diag '10'x
RELPAGE
375*



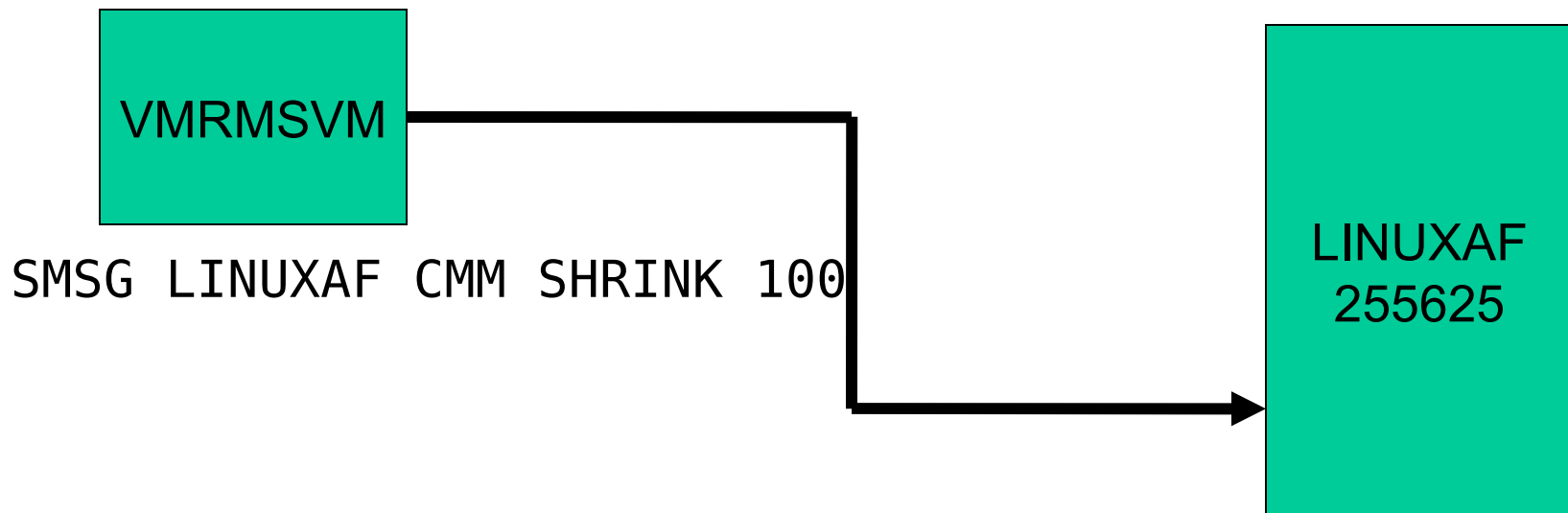
Page SHRINK: machine is now
255625 pages

after
RELPAGE
375

LINUXAF
255625

Next: Page SHRINK 100 pages

*Notify via CP SMSG
command*



Page SHRINK: can *increase by 275* page (375-100)

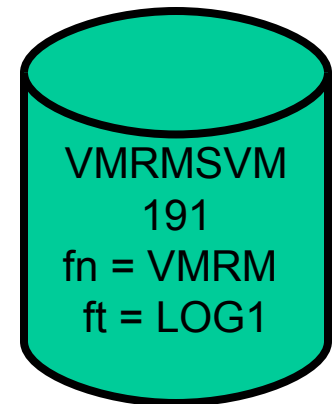
Allows the guest to
reclaim some of the
storage previously
released.

LINUXAF
255900

VMRMSVM logging -startup

```
2009-02-25 12:53:38 ServExe  Entry
-----
2009-02-25 12:53:38 ServExe  MSG          IRMSER0022I VM
Resource Manager Service Virtual Machine initialization
started
2009-02-25 12:53:38 ServExe  PCfg          VMRM CONFIG A1
2/25/09 7:54:41
2009-02-25 12:53:38 ServExe  InitEnv       MONITOR EVENT ACTIVE
BLOCK          500          P
ARTITION       8192

2009-02-25 12:53:38 ServExe  InitEnv       MONITOR DCSS NAME -
MONDCSS
2009-02-25 12:53:38 ServExe  InitEnv       CONFIGURATION SIZE
50 LIMIT
2009-02-25 12:53:38 ServExe  MSG          IRMSER0023I VM
Resource Manager Service Virtual Machine initialization
complete. Proceeding to connect to Monitor.
```



VMRMSVM logging - termination

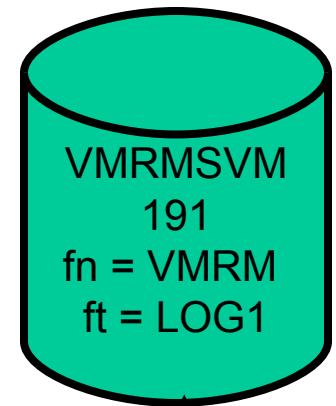
```
2009-02-25 12:53:38 MonRexx Entry MonIntCtr= 1 ,
Record= ENDR C3CD2331A780FC
80 , Processing this record at 25 Feb 2009 12:53:38

2009-02-25 12:54:06 MonRexx Entry MonIntCtr= 2 ,
Record= ENDR C3CD234C7CBE0A
:
80 , Processing this record at 25 Feb 2009 12:54:06

2009-02-25 12:54:28 MonExec Exit IRMMON0026I VM
Resource Manager processing of monitor records ended. Pipe
RC= 0

2009-02-25 12:54:28 ServExe MSG IRMSER0012I VM
Resource Manager Service Virtual Machine shutdown in
progress

2009-02-25 12:54:28 ServExe MSG IRMSER0027I VM
Resource Manager Service Virtual Machine shutdown complete
```



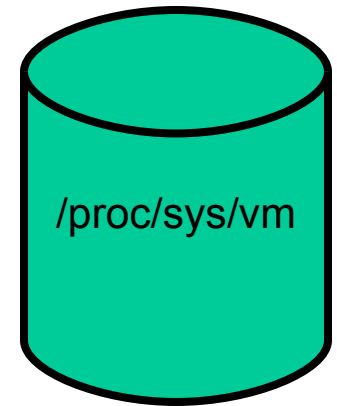
*Will close after
10,000 records.
Keeps 1 copy
around as
VMRM LOG2.*

Checking how many pages are participating

```
# cat  
/proc/sys/vm/cmm_pages
```

69362

in a 640m vm



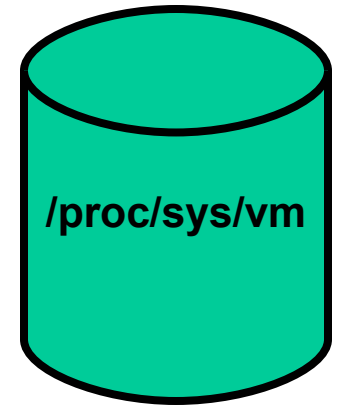
$69362/256 = 271 \rightarrow 271\text{Mb release}$

SHRINK values as reported in the VM RMSVM log file

| | | | | | |
|----------|--------|-------|--------------|------------|--------|
| 21:19:07 | MonCMM | CPCMD | SMSG LINUXA8 | CMM SHRINK | 37987 |
| 21:19:07 | MonCMM | CPCMD | SMSG LINUXA7 | CMM SHRINK | 135687 |
| 21:19:07 | MonCMM | CPCMD | SMSG LINUXA6 | CMM SHRINK | 90216 |
| 21:19:07 | MonCMM | CPCMD | SMSG LINUXA4 | CMM SHRINK | 48021 |
| 21:19:07 | MonCMM | CPCMD | SMSG LINUXA3 | CMM SHRINK | 120483 |
| 21:20:07 | MonCMM | CPCMD | SMSG LINUXA8 | CMM SHRINK | 69870 |
| 21:20:07 | MonCMM | CPCMD | SMSG LINUXA7 | CMM SHRINK | 138528 |
| 21:20:07 | MonCMM | CPCMD | SMSG LINUXA6 | CMM SHRINK | 110921 |
| 21:20:07 | MonCMM | CPCMD | SMSG LINUXA4 | CMM SHRINK | 90267 |
| 21:20:07 | MonCMM | CPCMD | SMSG LINUXA3 | CMM SHRINK | 128181 |

Checking how many pages are participating

```
# cat /proc/sys/vm/cmm_pages
114687
# cat /proc/sys/vm/cmm_timed_pages
0
# cat /proc/sys/vm/cmm_timeout
0 0
```



CMMA – VM and Linux levels

- z/VM 5.3 plus APAR VM64265 and APAR VM64297
- SLES10 SP1 update kernel 2.6.16.53-0.18
- Redhat – not available

CMMA – instruction level communication

- Uses storage key to describe page contents
- ESSA instruction

| Bit Desc. | Meaning |
|----------------------|---|
| Stable | Guest cannot recreate contents |
| Unused | Unused |
| Volatile | Useful content but may be discarded. Discard fault presented to guest |
| Potentially volatile | Dirty = CP handles as Stable Not dirty = CP handles as volatile |
| | |

CMMA – instruction level communication

- CP investigates the bit settings:
 - Possibly steal unused, volatile, not dirty potentially volatile pages without necessarily having to page out contents.
 - CP can use clean disk cache pages, and if Linux needs it back, CP will reflect a discard interrupt.
 - Linux marks a page for removal CP may reuse it without having to page out.
 - Assist provided (Host Page-Management Assist) to let guest reclaim discard page without CP interception (remains runnable).

System and Linux mechanics

- MEMASSIST must be on for system and virtual machine.

```
cp query memassist
```

```
ALL USERS SET - ON
```

| USER | SETTING | STATUS |
|-------|---------|----------|
| MAINT | ON | INACTIVE |

```
cp query memassist linuxa1
```

```
ALL USERS SET - ON
```

| USER | SETTING | STATUS |
|---------|---------|--------|
| LINUXA1 | ON | ACTIVE |

System and Linux mechanics

- Kernel parameter is `cmma=on`

```
# dmesg | grep cmma
```

```
Kernel command line: root=/dev/ram0 init=/linuxrc rw  
barrier=off selinux=0 TERM=dumb elevator=cfq cmma=on  
BOOT_IMAGE=2
```

CP TRACE ESSA STEP 5

*Tracing the ESSA instruction in a
Linux virtual machine with class G
TRACE command*

CP TRACE ESSA STEP 5

```
-> 000000000001C00FA'  ESSA  B9AB2001  0000000000000000
                                     00000000159AF000  CC 2
-> 000000000001BD4EC'  ESSA  B9AB1051  0000000000000004
                                     00000000159AF000  CC 2
-> 000000000001BD4EC'  ESSA  B9AB1051  0000000000000004
                                     0000000015AAA000  CC 2
-> 0000000000020A596'  ESSA  B9AB6021  0000000000000000
```

Case Study

- When running WAS “idle” Linux machines remain in Q3 forever.
- Using resource needlessly, causing storage overcrowding in the high rent district.
 - Never drops down to dormant list
- Attempted to duplicate problem in test lpar.
 - *However machines do not sit in Q3 – but they still work through queues even when “idle”*

Production System Queue Reports: 03:15 – 03:30

Velocity Software report ESAUSRQ

```

1Report: ESAUSRQ      User Queue and Load Analysis      Velocit
Monitor initialized: 02/25/09 at 03:00:00 on 2094 serial ABCDE      First r
-----
          <-----User Load----->          <-----Average Numbe
UserID   Logged  Non-          Disc- Total  Tran <-----Dispatch List----
/Class      on  Idle  Active  conn  InQue  /min  Q0    Q1    Q2    Q3
-----
03:30:00  87.0    .    70.3    .    55.9  781    0   10.5   6.3  39.1
Hi-Freq:  87.1   70    70.3    84   55.6  780    0.2  10.6   5.6  39.2

```

Same results at 3 a.m. ...

Production System Queue Reports

15:15 – 15:30

Velocity Software report ESAUSRQ

```

1Report: ESAUSRQ      User Queue and Load Analysis      Velocity
Monitor initialized: 02/25/09 at 15:00:00 on 2094 serial ABCDE      First re
-----
              <-----User Load----->              <-----Average Number
UserID   Logged  Non-          Disc- Total  Tran <-----Dispatch List-----
/Class      on  Idle  Active  conn  InQue  /min  Q0   Q1   Q2   Q3  L
-----
15:30:00  87.0    .    70.3    .  56.0  725    0  10.7  5.3  40.0
Hi-Freq:  87.1   70   70.3   84  55.6  725   0.4  9.5  5.7  39.9
0 ***Key User Analysis ***

```

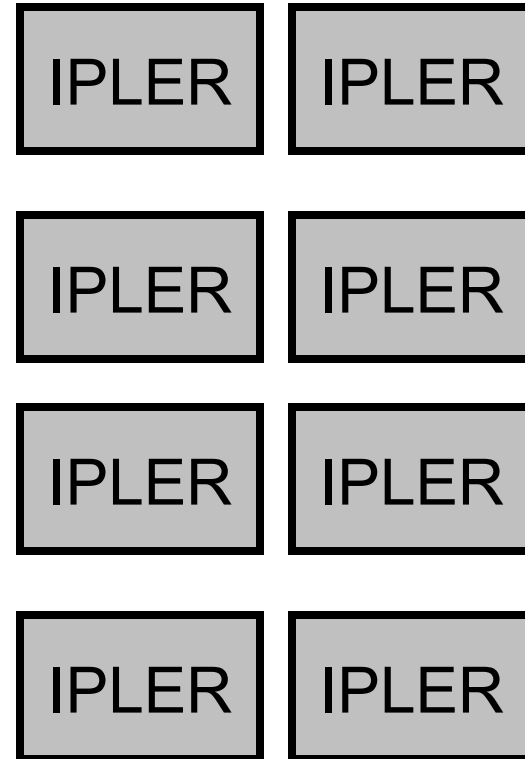
Same results at 3 a.m. as at 3 p.m.

Test case: CP Storage: 2.5G with XSTORE .5G z9 with two IFLs uncapped LPAR

1. WAS started; each machine 1.5G



2. IPL LINUX; each machine 1G



Tried this approach

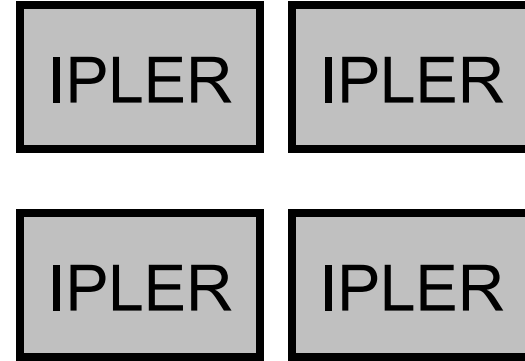
- Over commitment of 10:1 and higher
 - Inquiring minds want to know!
 - Done in a test LPAR
- Results were just not good
- Certain runs:
 - Caused thrashing
 - Exceptionally high CP overhead – CP tries to keep all vm's happy ends up punishing all!
 - Elist formation
 - Severe memory resource shortage

Trying for the sweet spot: CP Storage: 2.5G with XSTORE .5G z9 with two IFLs uncapped LPAR

1. WAS started; each machine 640M



2. IPL LINUX; each machine 512M



Results

- Caveat: results are mine only based on limited circumstance testing.
- Caused extreme memory stress during most tests.
- Overcommitment of 10:1 didn't work so well.
 - So what's the right number: between 1 and 10...
 - Around 3 – 4? 5? ... 6?
 - And cmm and cmma can help with overall storage management with careful management
- By no means formal tests.
- Will continue to evaluate

Comments

- **The VMRSVM “kick in” determined by “black box” internal values; no control.**
- **Maybe it was the nature of the tests but...**
 - **External setting of low, medium or high relpage processing would be nice.**
 - **Follow suggestions for using CMM with non-production workloads.**
 - **CMM-1 and CMMA are not “set it and forget it”**
 - **Requires a performance monitor!**
 - **Used Velocity products, CP, and linux commands.**
- **Nonetheless CMM-1 and CMMA are reasonable tools in the right hands.**

Perceptions and Recommendations

- After the VM RMSVM has instructed servers to give up a lot of pages:
 - Simple tasks in those machines had elongated response times
 - Attempts to ssh
 - 3270 logon
- Recommendation: Do not place production Linux servers in VM RMSVM target list.
 - Only have VM RMSVM manage test, development, sandbox Linux servers.
- Use CMM-1 in LPARs that have mixed production and non production workloads.

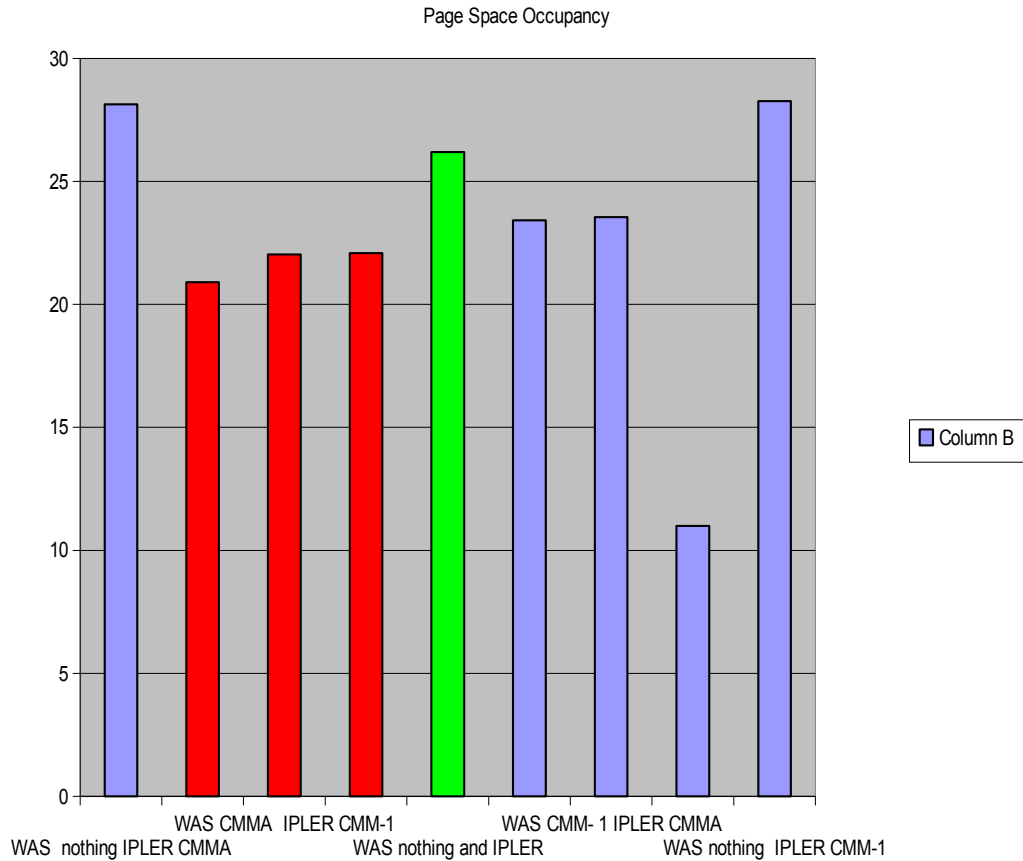
**Test suites:
6 WAS at 640M 4 IPLERS at 512M**

9 tests performed

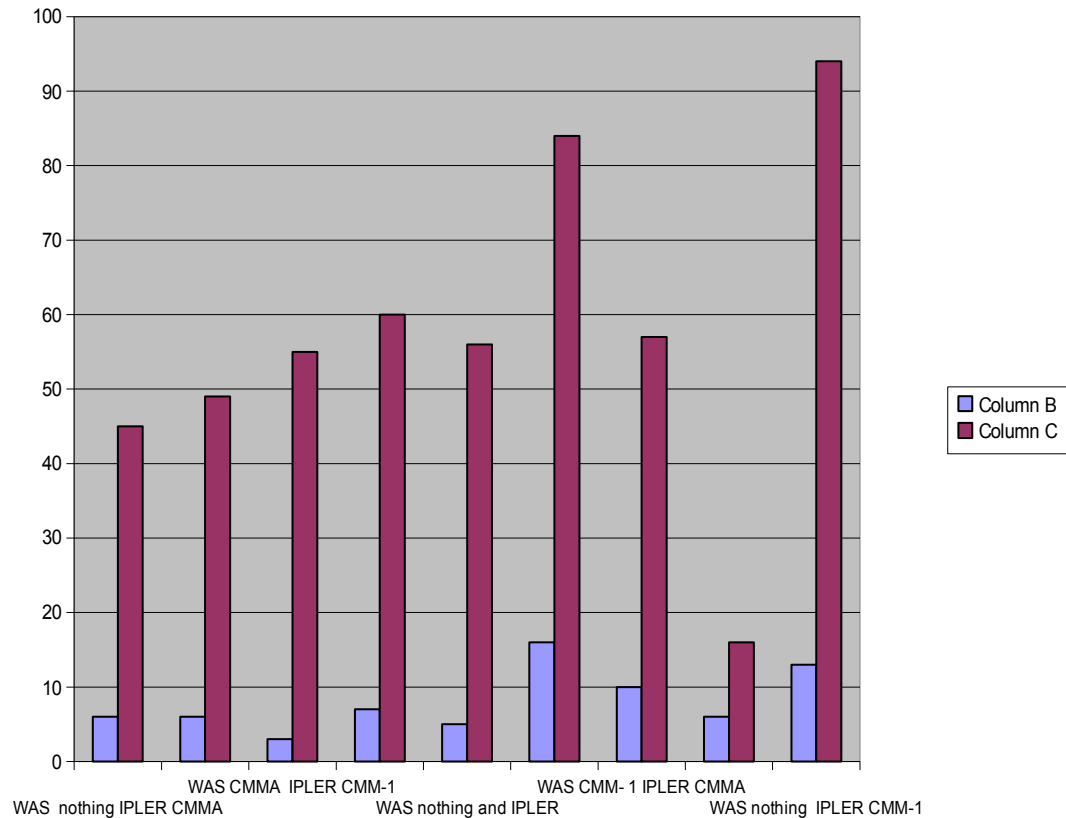
Test Case Matrix

| | IPL-er No CMM | IPL-er CMM-1 | IPL-er CMMA |
|---------------|------------------|-----------------|----------------|
| WAS No CMM | 1 | 2 | 3 |
| WAS CMM-1 | 4 | 5 | 6 |
| WAS CMMA | 7 | 8 | 9 |

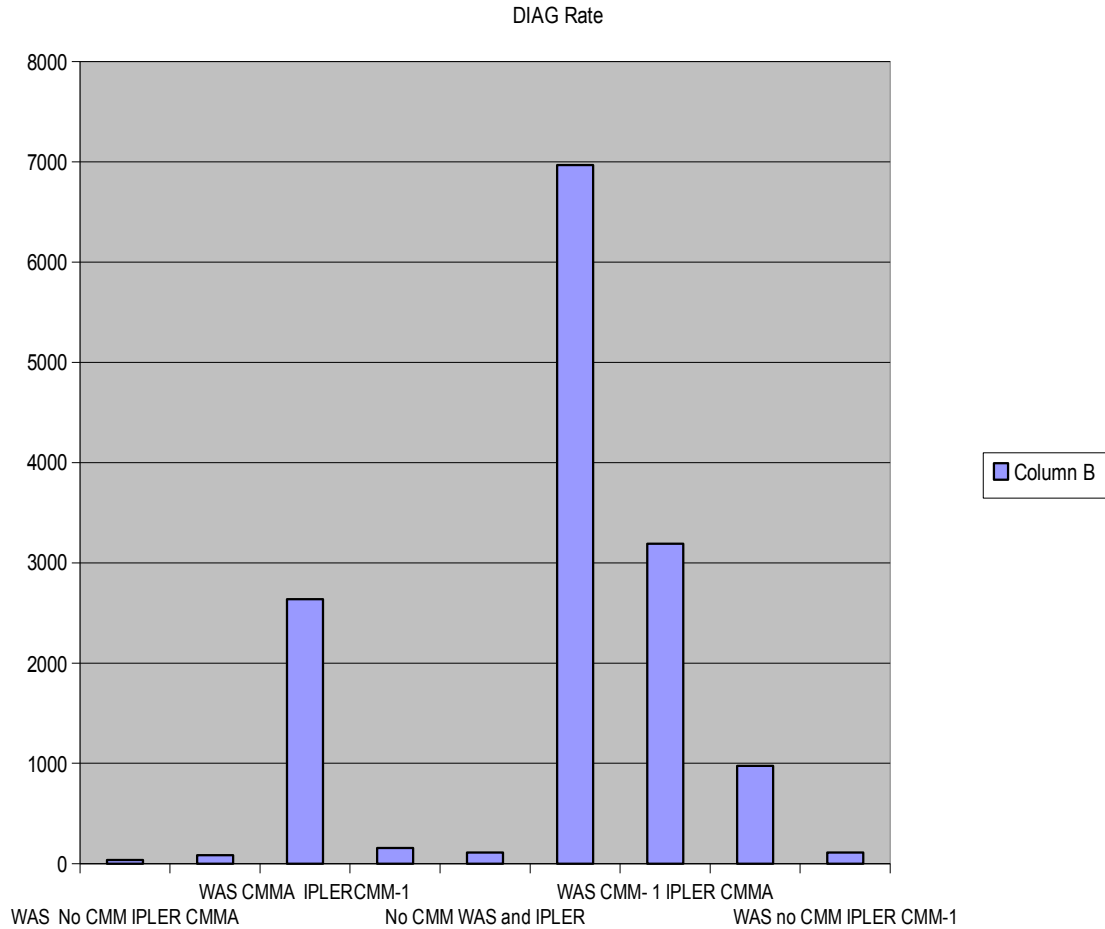
Paging Disk Occupancy



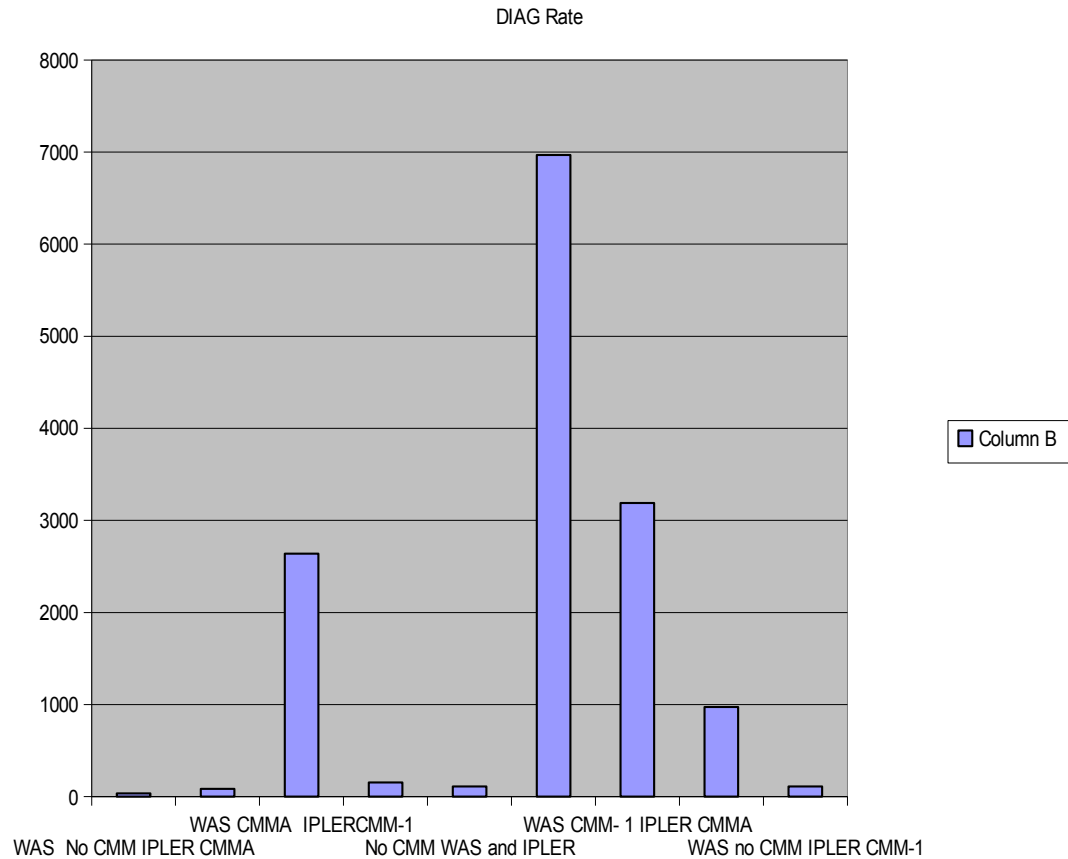
User CPU and CP Overhead



Working Set Sizes



DIAG Rate



| Test | Page read sec | Page Write sec |
|-------------------------------|------------------------------|-------------------------------|
| WAS no CMM IPLER CMMA | 788 | 1217 |
| WAS CMMA IPLER CMMA | 421 | 1717 |
| WAS CMMA IPLER CMM-1 | 42 | 208 |
| WAS CMMA IPLER No CMM | 782 | 1927 |
| no CMM WAS and IPLER | 242 | 1038 |
| WAS CMM-1 IPLER no CMM | 2989 | 3220 |
| WAS CMM- 1 IPLER CMMA | 1847 | 2172 |
| WAS CMM- 1 IPLER CMMA | 2004 | 1234 |
| WAS no CMM IPLER CMM-1 | 3004 | 5112 |

Late breaking news....

- IBM will depreciate CMM-A in future code releases
 - Had difficulty getting CMM-A accepted by the Linux kernel developers
 - Parts of it may reappear in the future

Thank you to

- Barton Robinson
- Dave Jones
- Dominic Coulombe