Sharing the Love of z/VM 620 Clustering Solutions

VM WORKSHOP 2013

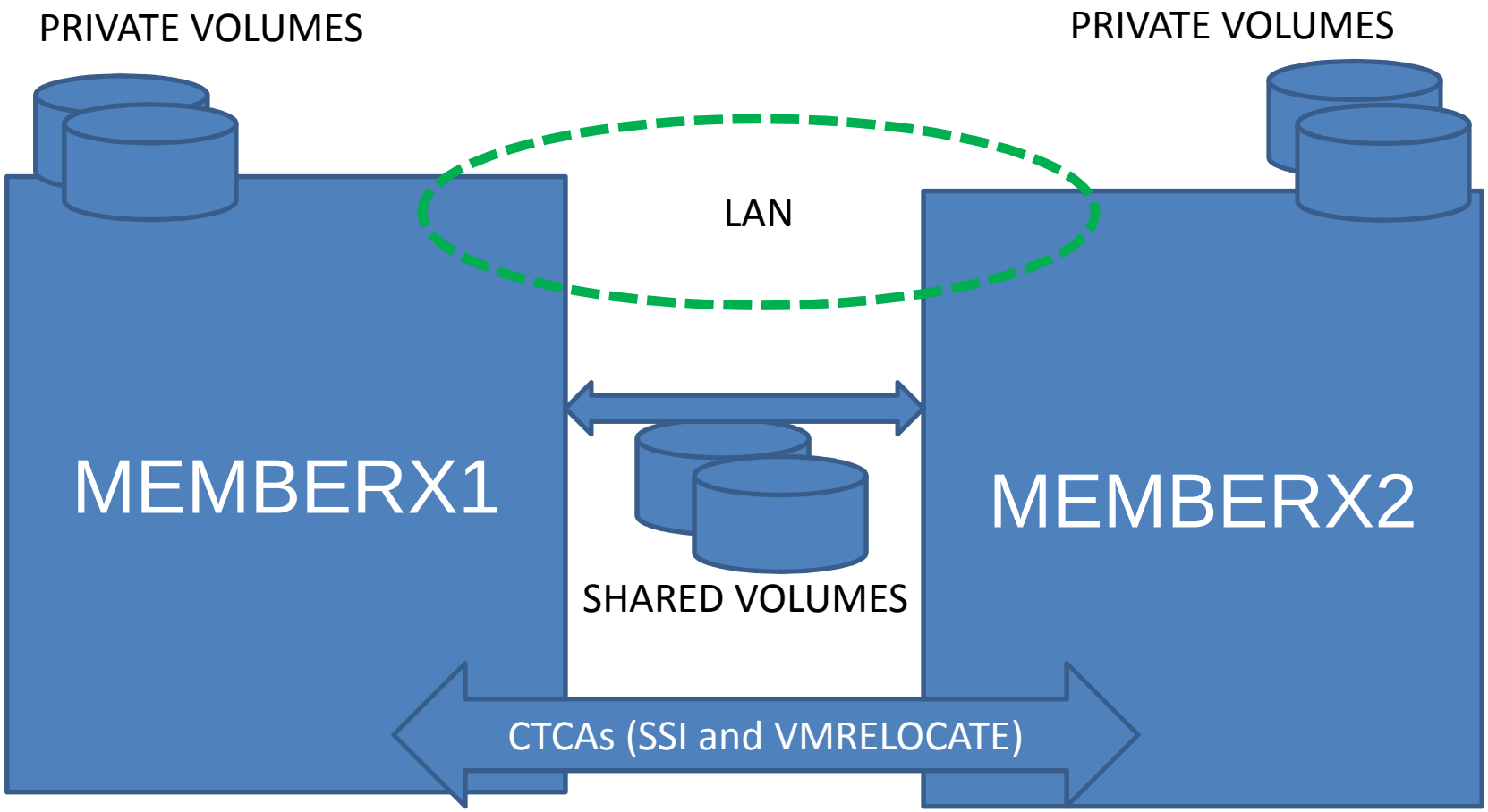
Prepared by: David Kreuter

Presented by: Dave Jones

Today’s Presentation As Advertised

- By the time we're gathered in the heat of the Workshop many of us will be running z/VM 620. In this presentation David will discuss the usage of z/VM 620 at his clients. As usual David's technical drill down will be replete with information on clustering system design, LPAR setup, tool smithing, networking, and some slick methods of dealing with multiple clusters.

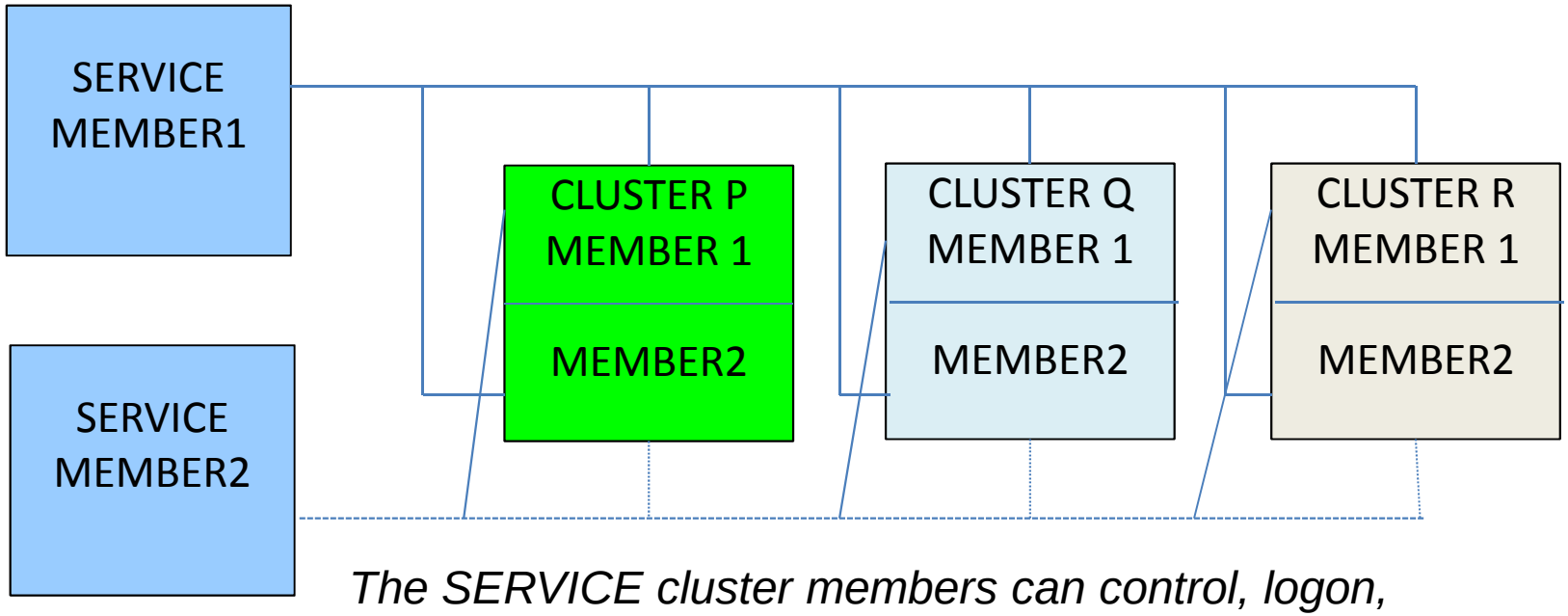
A Client's Typical SSI Environment:



Some Lessons Learned

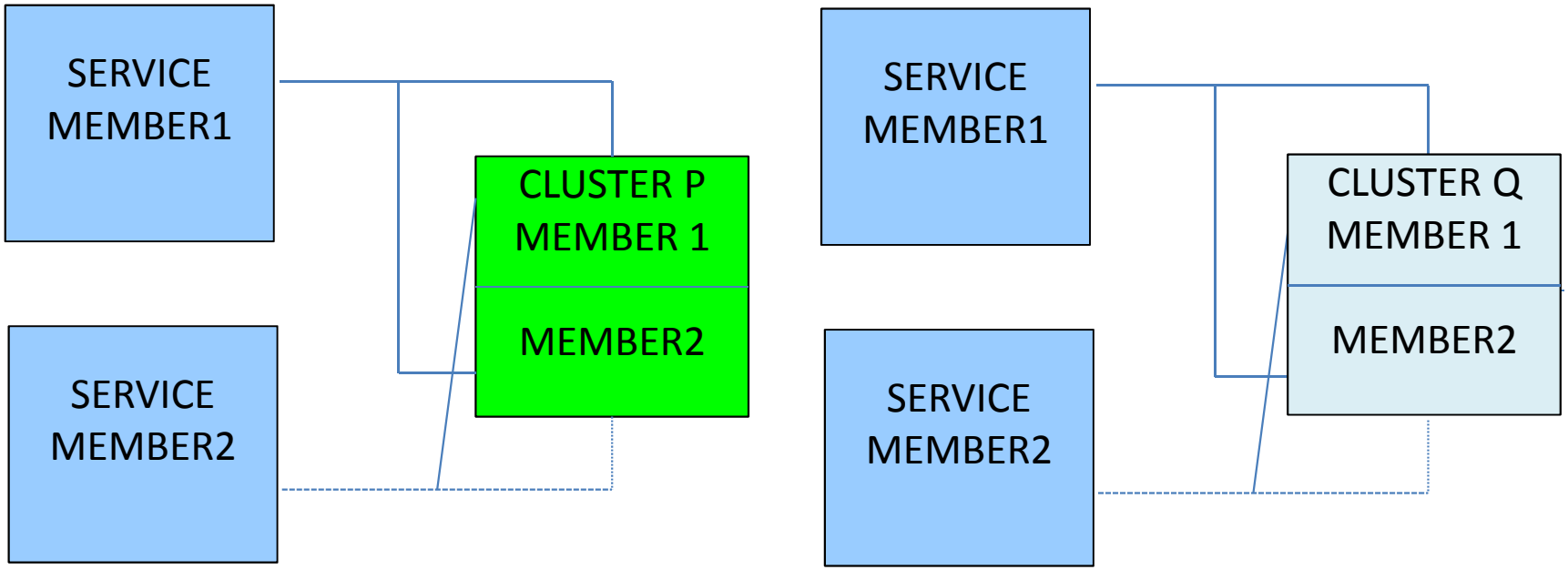
- IDENTITY machines and SFS
- Mapping minidisks
- DIRMAINT usage
- RACF usage
- VMRELOCATE ease of use
- Lots of work managing multiple LPARs!

Clustering System Design



The SERVICE cluster members can control, logon, file transfer, manage, do directory and RACF work on any MEMBER in any cluster. If in the same CEC connected via hipersocket. Work is usually done from MEMBER1. MEMBER2 can assume any responsibility.

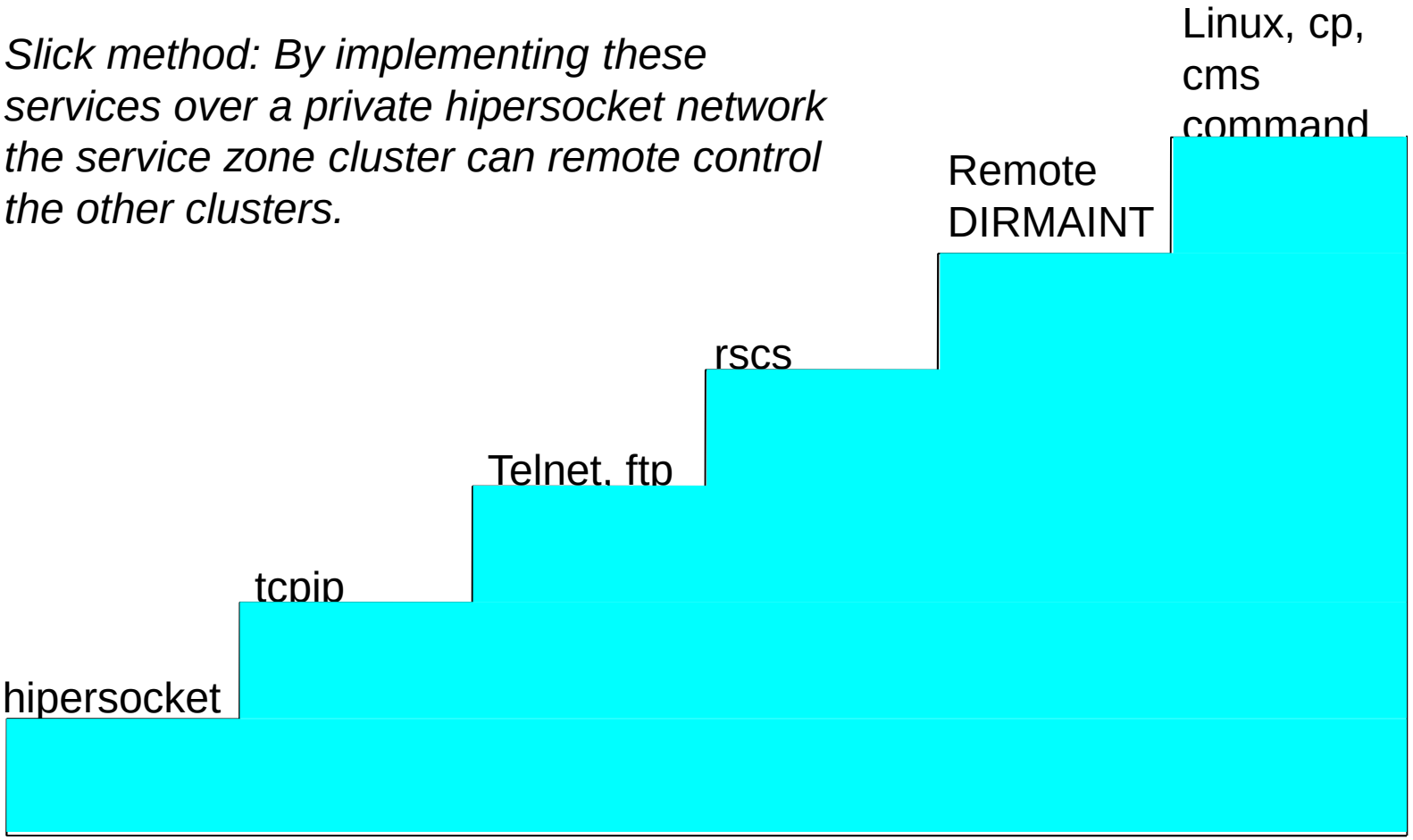
Clustering System Design



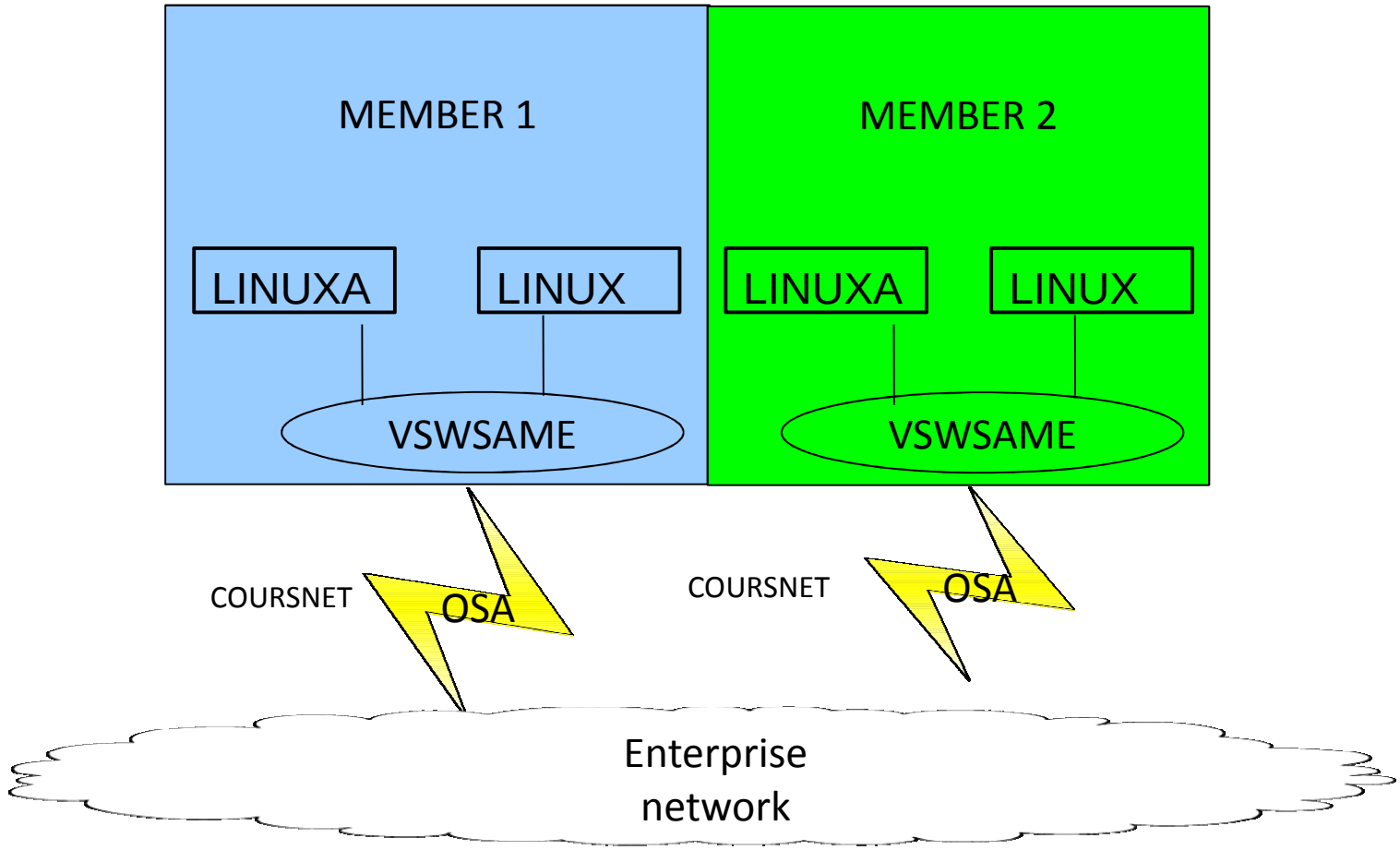
The SERVICE cluster can control any other cluster but the different clusters are blind to each other.

Service Zone Step Up

Slick method: By implementing these services over a private hipersocket network the service zone cluster can remote control the other clusters.

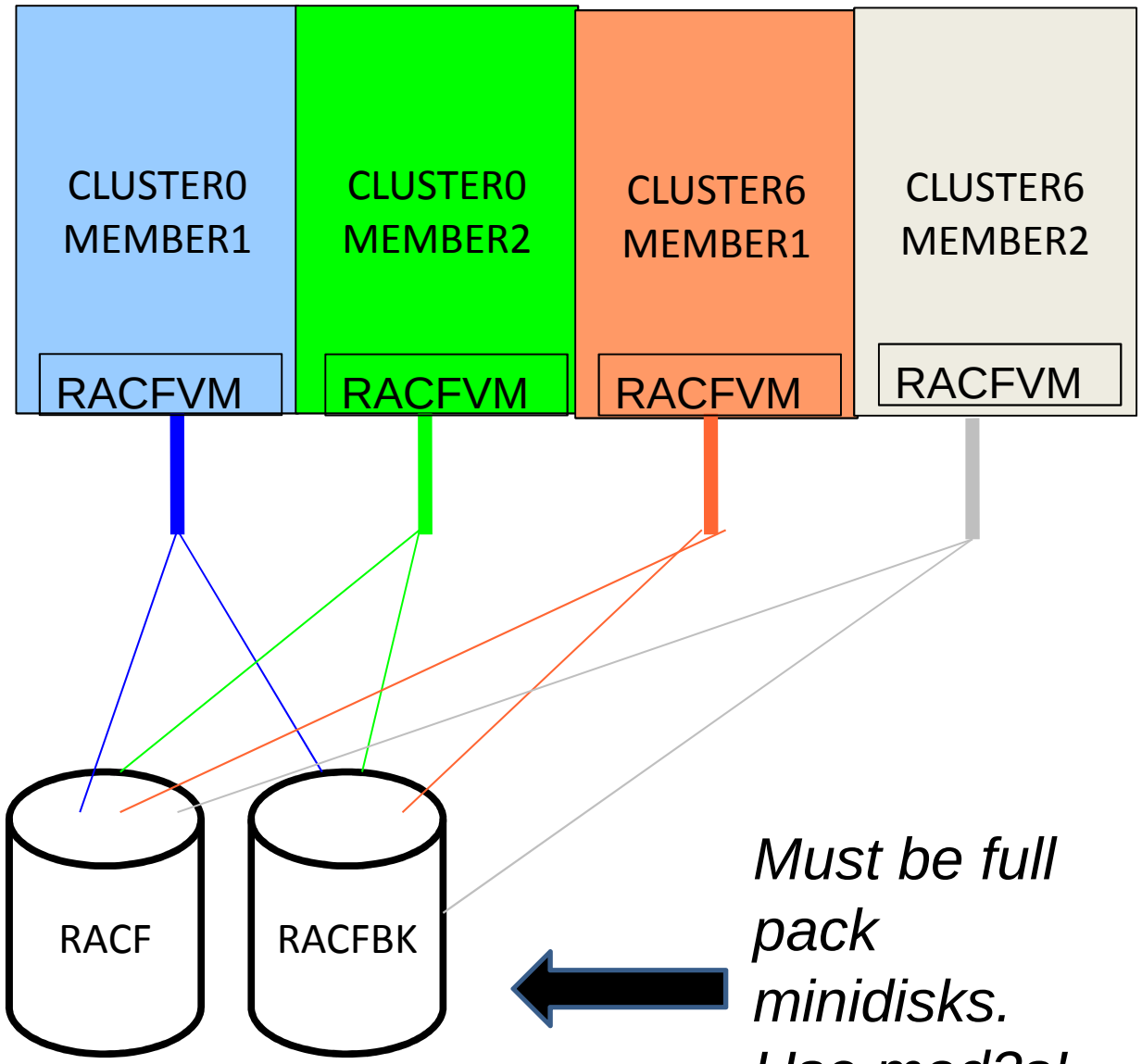


In zvm620 the VSWSITCHes should have the same name on every member. The OSAs must have the EQIDs.



RACF DB Sharing

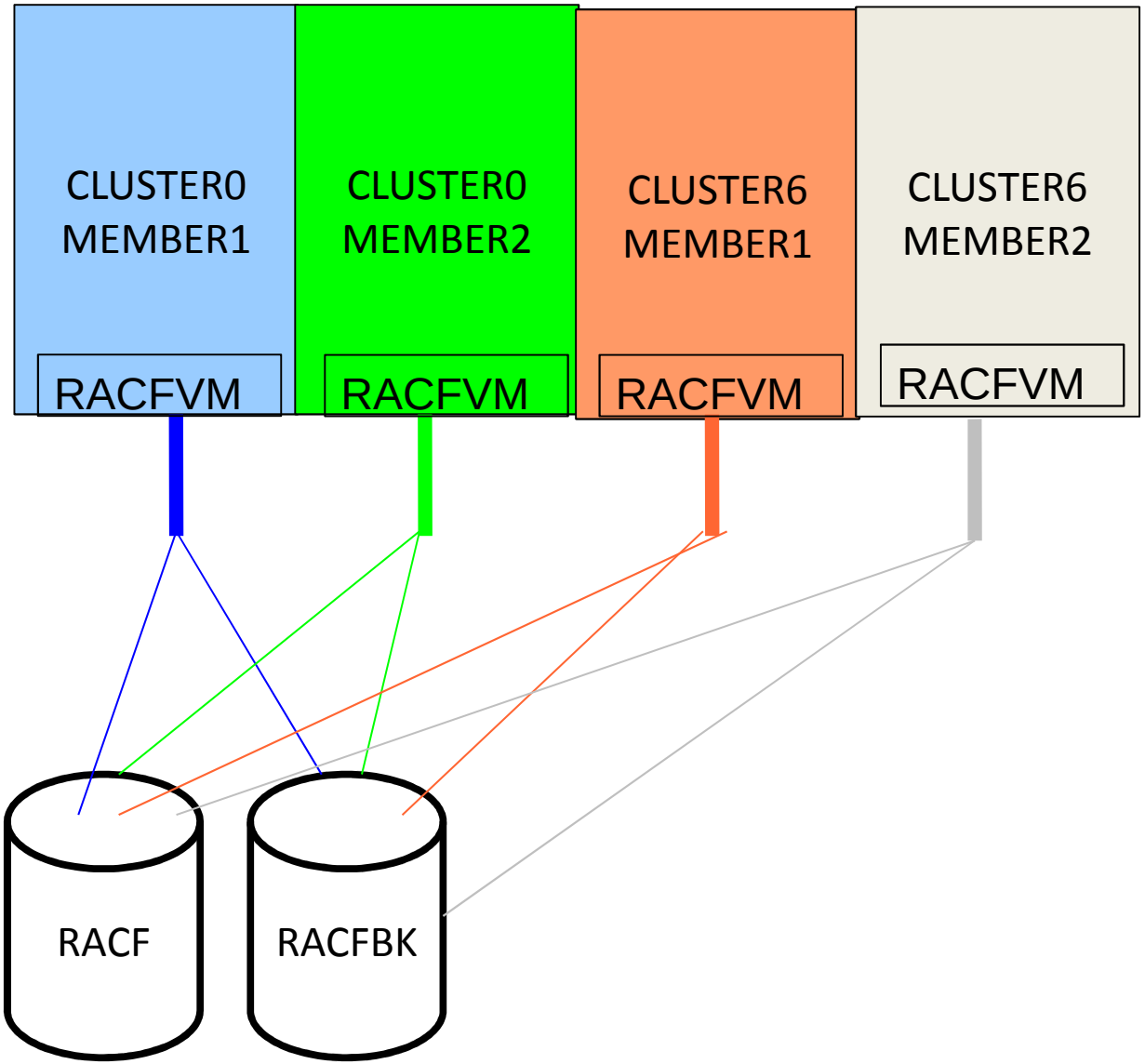
The RACF database is shared by all Clusters. The DASD are marked as shared in the SYSTEM CONFIG. in LPARs. All RACF administration tasks may be performed in the SERVICE zone. In ZVM620 the RACF DB must be on a full pack minidisk.



Must be full pack minidisks. Use mod3s!

RACF Lessons Learned

- Do the occasional RAC SETROPTS REFRESH for RACLISTed classes.
- Do NOT hook up the RACF DBs from the 2nd level systems.
  - Keep as sand box.
- Moving profiles, groups, users from 610 to 620 is a chore.
  - Authored local tools.



LPAR Setup Lessons Learned

- Share everything
- Tailor I/O in the z/VM Member in SYSTEM CONFIG.
  - Mark as OFFLINE or NOT_ACCEPTED devices you do not need.
- z/OS is the owner.
- Dynamic I/O is great.
- Keep some memory reserved.
- Learn how to use the URM.

```
/**********************/
/* Status of Devices */
/**********************/

VMRSRV01: Devices,
Online_at_IPL ,
0000-FFFF,
SHARED 6706,
SHARED 670F,
Offline_at_IPL,
2C00-2C0F,
2D00-2D0F,
3000-5FFF,
Sensed 0000-FFFF
```

Locally Authored Tools

- Timer based console closing of CMS and Linux servers.
- A hot reader collector of console logs.
- Alerting EXEC that sends out alert emails.
- Directory synchronization tools.
- Linux Minidisk Backup Tools.
- Cross System Commands

For z/OS 'cause all you have are hammers! … And everything is a nail.

In z/VM the surgical precision of the CMS tools is great!

Locally Authored Tools: Hints and Tips

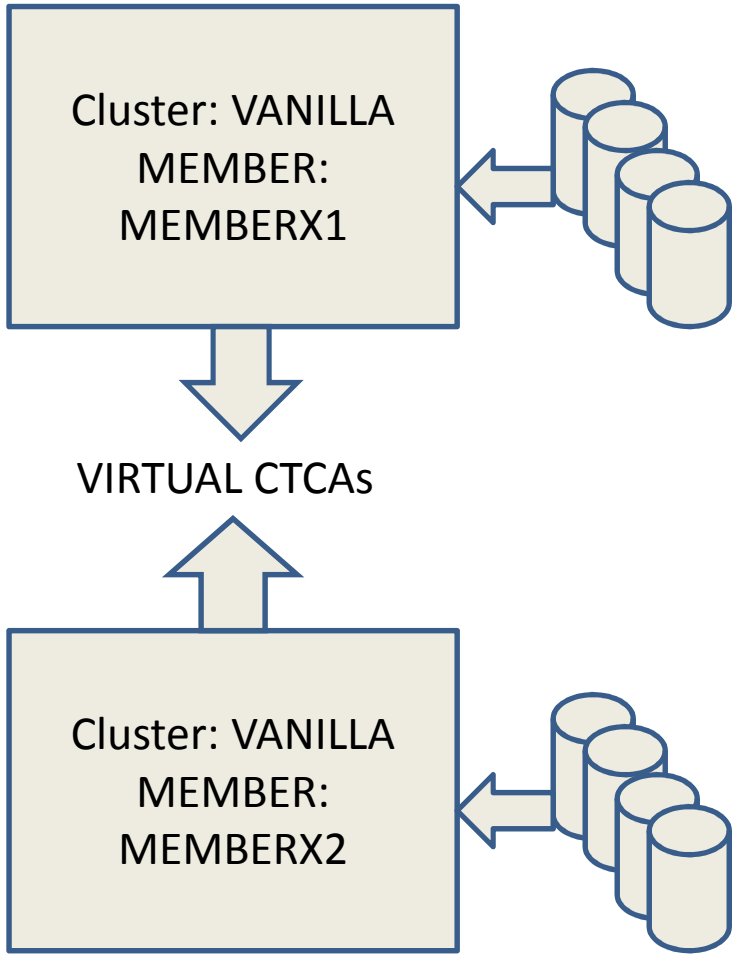
- Install in your Golden Cluster Image
  - Otherwise you will have to manually install on many VM LPARs. No fun.
- Externalize as many names and values as possible lest you be maintaining many code images.
  - It becomes private code.
- Write a routine that checks disk or SFS space and ALERT via email on thresholds.
  - Call this from any tool that WRITEs files.
- Document in hard copy.

For z/OS 'cause all you have are hammers! … And everything is a nail.

In z/VM the surgical precision of the CMS tools is great!

Cluster Generation Methods

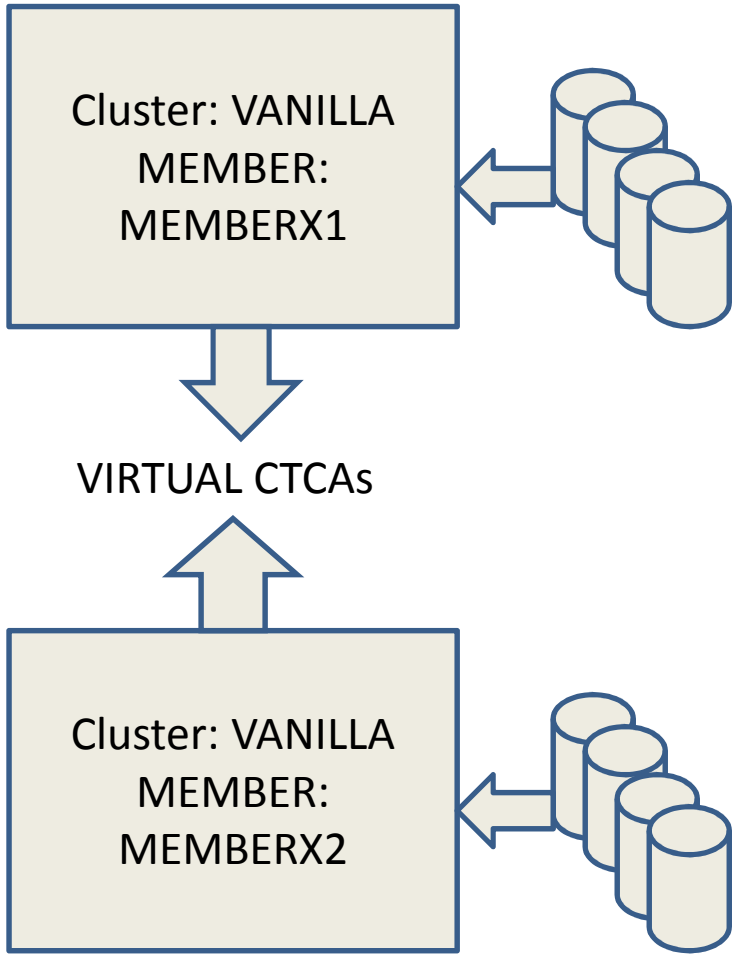
- Two second level machines used for new cluster generation.
- All products installed and serviced.
- Extensive Tailoring.
- Local tools and machines installed.
- Test this engine!
- *Each member has a SYSRES with direct cylinders.*



Cluster Generation Steps

- Second level SSI systems glued together with virtual CTCAs:

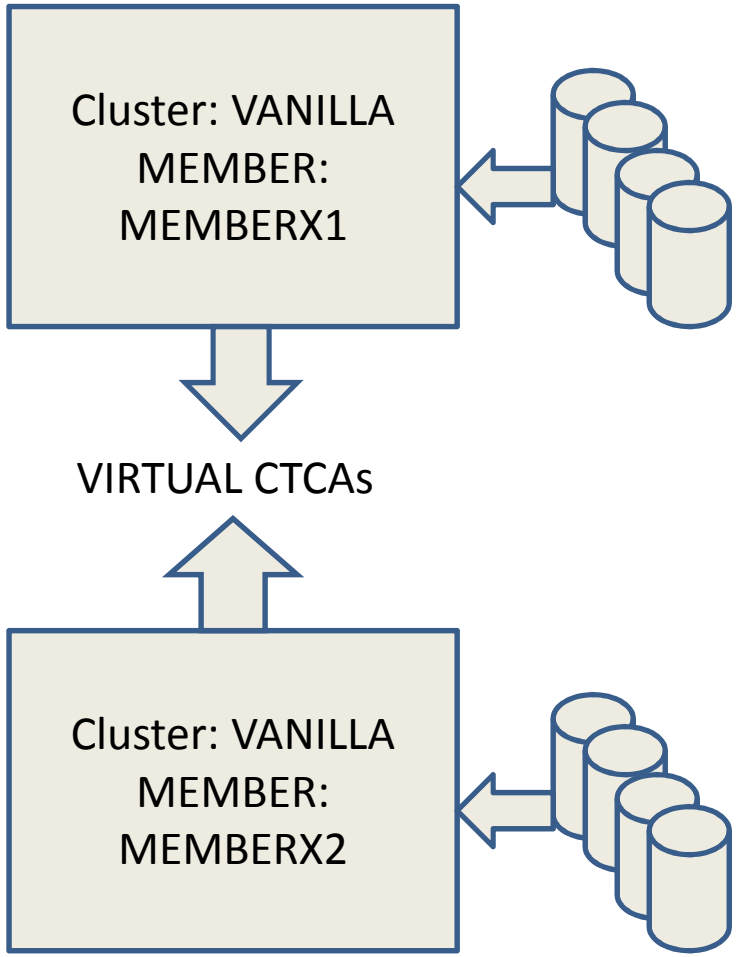
1. Install IBM code, RSUs, and reach ahead PTFs.
2. Tailor IBM “stuff”
3. Install local tools.
4. Test
5. Do more testing
6. FLASHCOPY or DDR to new systems



Cluster Generation New System

- New volumes starting on cylinder 0

1. Using CPFMTXA reLabel, reAllocate and place ownership on required volumes.
2. Run FORMSSI on the new common volume.
3. On the PMAINT 41D tailor the VM SYSPINV file
   - You will be sorry if you do not do this! SERVICE and PUT2PROD will be most unhappy!

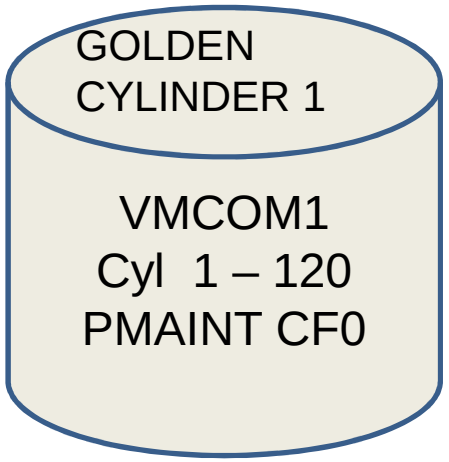


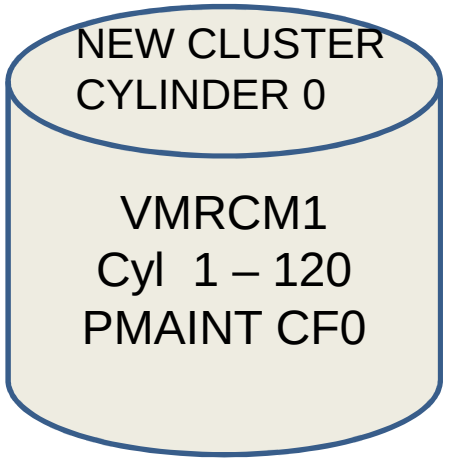
Cluster Generation: DIRECTORY and the Mapped MDISK trick

- New cluster volumes starting on cylinder 0
- Golden image volumes start on cylinder 1

1. Take a copy of the 2nd level directory to the first level.
2. Using XEDIT change all volsers on MDISK statements.
   1. Change the DIRECTORY statement too!
3. Slam this directory into your new system!
   1. RC = 5 desirable – it means you loaded a directory but not on the production system.
   2. Copy to the new DIRMAINT 1DF as USER INPUT
   3. Erase USER DIRECT on new DIRMAINT 1DF
4. Create a map of all MDISKs on the new volumes
   1. NOT a map of your golden 2nd level images.
   2. IS a MAP of MDISKs on NEW system.

- *EXAMPLE: shows the PMAINT CF0 minidisk from Golden (VMCOM1) and NEW CLUSTER (VMRCM1)*



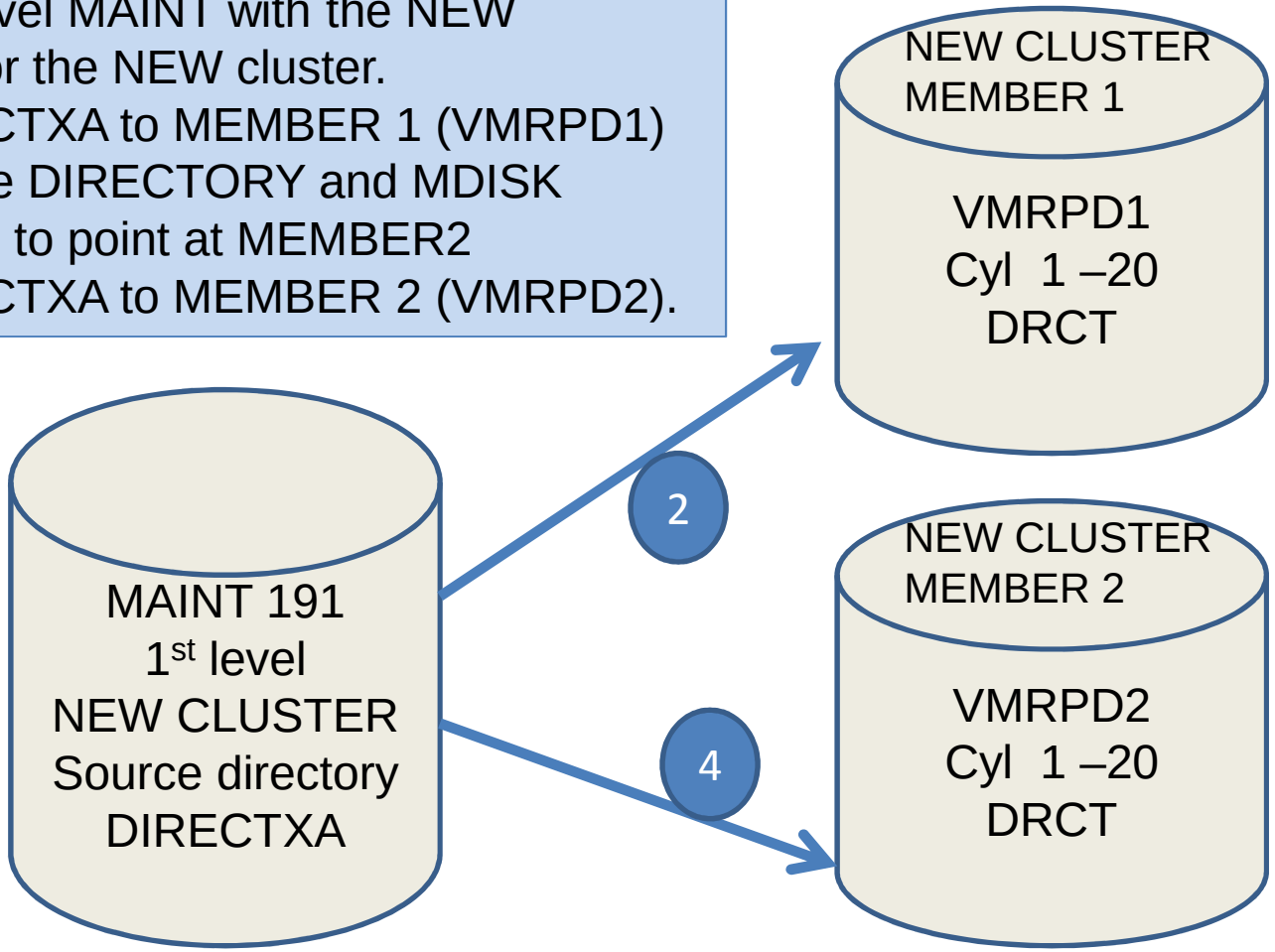


PMAINT MDISK 0CF0 3390 001 120 VMCOM1

Cluster Generation: DIRECTORY LOAD

DIRECTORY load:

1. Use first level MAINT with the NEW directory for the NEW cluster.
2. Run DIRECTXA to MEMBER 1 (VMRPD1)
3. Change the DIRECTORY and MDISK statements to point at MEMBER2
4. Run DIRECTXA to MEMBER 2 (VMRPD2).



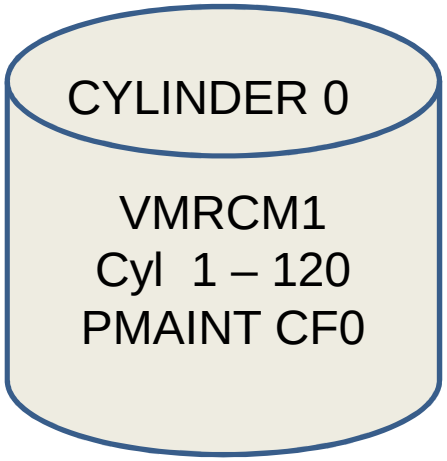
Cluster Generation: the Mapped MDISK trick

- The DIRFAMOS EXEC uses a source directory as input and produces reports with the USER or IDENTITY as the first field. Sample:

```
PMAINT MDISK 0CF0 3390 001 120 VMRCM1
```

- Very useful for mapping minidisks.
- Used as well to compare directory entries.

```
q da vmrcm1
DASD 5F00 VMRCM1
Ready; T=0.01/0.01 01:37:52
att 5f00 system
DASD 5F00 ATTACHED TO SYSTEM VMRCM1
Ready; T=0.01/0.01 01:37:56
def mdisk fcf0 1 120 vmrcm1
DASD FCF0 DEFINED
Ready; T=0.01/0.01 01:38:04
```



DIRFAMOS – it's famous!

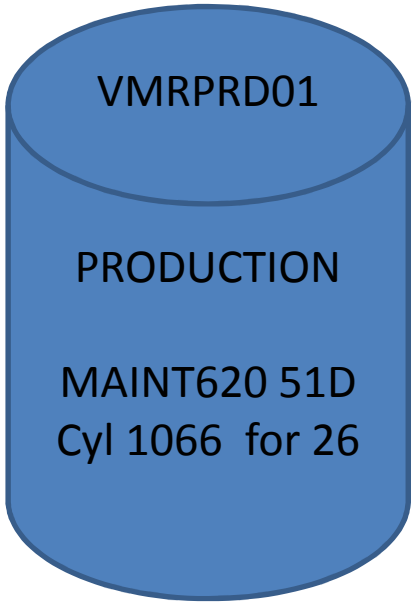
- Code available – just ask.
  - As is.
- Mostly PIPELINEs:
  - READ a source DIRECTORY
  - Place the USERID or IDENTITY as first field on very record.
  - Like a mini Data Base of the directory.
  - Great for mapping minidisk.
  - WRITE multiple reports.

```
PMAINT MDISK 0CF0 3390 001 120 VMRCM1
```

The Greatness of the Service Zone

Since the system packs are copies from the Golden Image it is necessary to update the member names in certain files. Many of these files live on MAINT620's 51D disk.

Minidisks from the other clusters are linked from the service zone. A "code blast" is done to these minidisks to make all the necessary changes. You do not have to logon to the other clusters!



Example of file contents:

```
VM       SYSSUF  D1
4OSASF40 SRVPROD D1
5684042J SRVPROD D1
6VMCMS20 SRVPROD D1
6VMCPR20 SRVPROD D1
6VMDVF20 SRVPROD D1
6VMHCD20 SRVPROD D1
6VMPTK20 SRVPROD D1
6VMTCP20 SRVPROD D1
```

Files that contain member names on the "D" disk

```
6VMTCP20 SRVPROD D1 contains:
:PTF.UK73639
:STAT.PUT2PROD.12/18/12.14:26:40.MAINT620.VMR200Q

PUT2PROD.12/18/12.14:19:48.MAINT620.VMR200P
BUILT.12/18/12.14:16:20.MAINT620
```

Do for Each Cluster's MAINT620 51D Minidisk

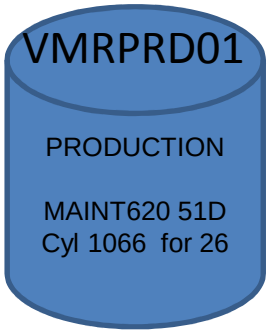
```
q da vmprl1
DASD 5F08 CP SYSTEM VMPRL1 0
Ready; T=0.01/0.01 13:28:47

pipe < vmrprod01 direct a
|all /MAINT620/ & /51D/
|cons
MDISK 0051 3390 1067 026 VG4RL1 MR PW
MAINT620 051D

link vmrprd01 51 51 mr
Ready; T=0.01/0.01 13:30:11

ac 51 d
Ready; T=0.01/0.01 13:30:14
```

This sequence of commands from the SERVICE zone LINKs to the MAPPED MINIDISK (The MAINT620 51d minidisk from the PROD cluster).



Run the "FINDB" EXEC to Change File Conterns

```
Findb vmrprd01 vmrprd02
Ready; T=0.01/0.01 13:29:47
```

This code run on the SERVICE zone changes the strings "vm200p" and "vm200q" to "vmrprd01" and "vmrprd02" on the mapped disk.

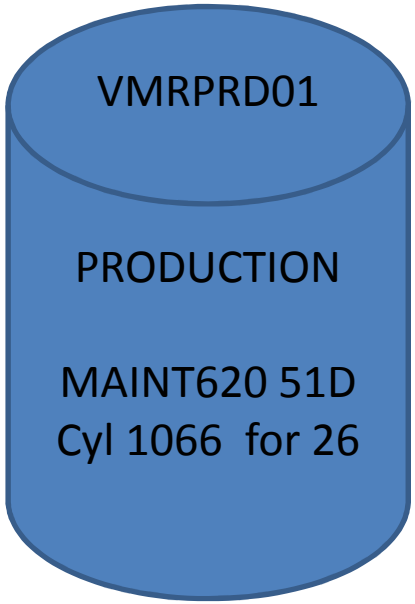
```
6VMTCP20 SRVPROD D1 contains:
:PTF.UK73639
:STAT.PUT2PROD.12/18/12.14:26:40.MAINT620.VMRPRD02
PUT2PROD.12/18/12.14:19:48.MAINT620.VMRPRD01
BUILT.12/18/12.14:16:20.MAINT620
```

Sequence of events "Code Blasted" to all clusters!

The Greatness of the Service Zone

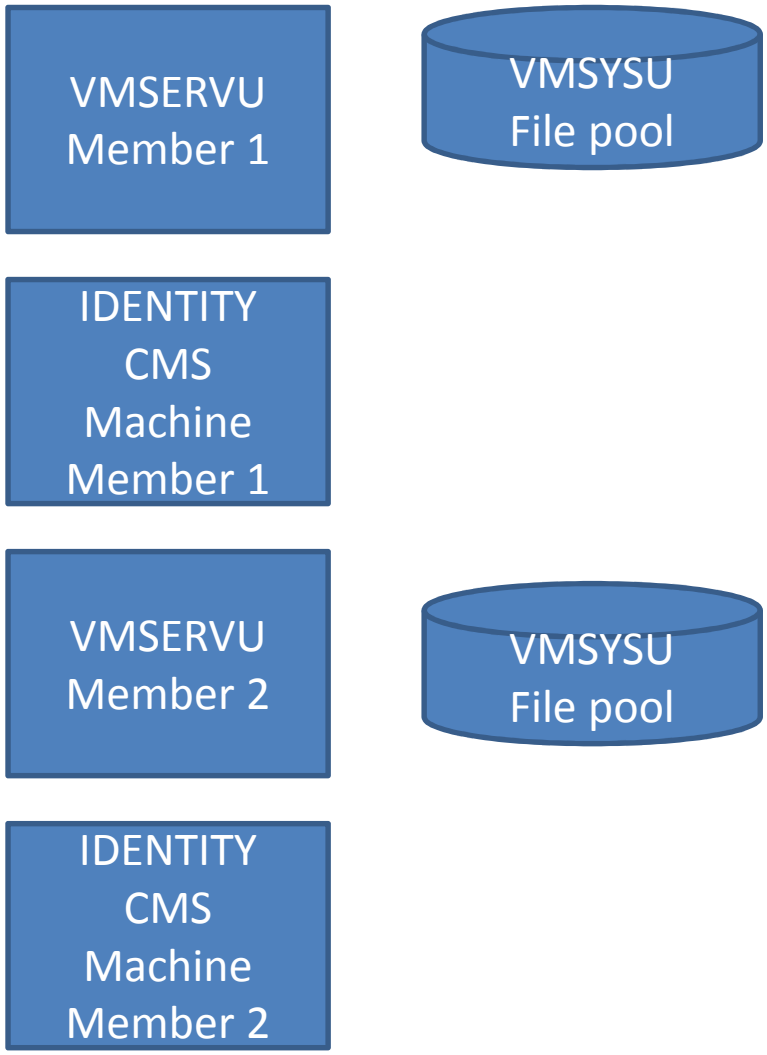
Since the system packs are copies from the Golden Image it is necessary to update the member names in certain files. Many of these files live on MAINT620's 51D disk.

Minidisks from the other clusters are linked from the service zone. A "code blast" is done to these minidisks to make all the necessary change. You do not have to logon to the other clusters!



IDENTITY Discovery

- For CMS usage IDENTITY defined machines work great with SFS space!
  - No SUBCONFIG entries required
  - No MDISK statements needed.
  - Really nice for local tools.
  - Boost up disk storage size of VMSYSU file pool.
  - Each member has unique VMSYSU file pool.

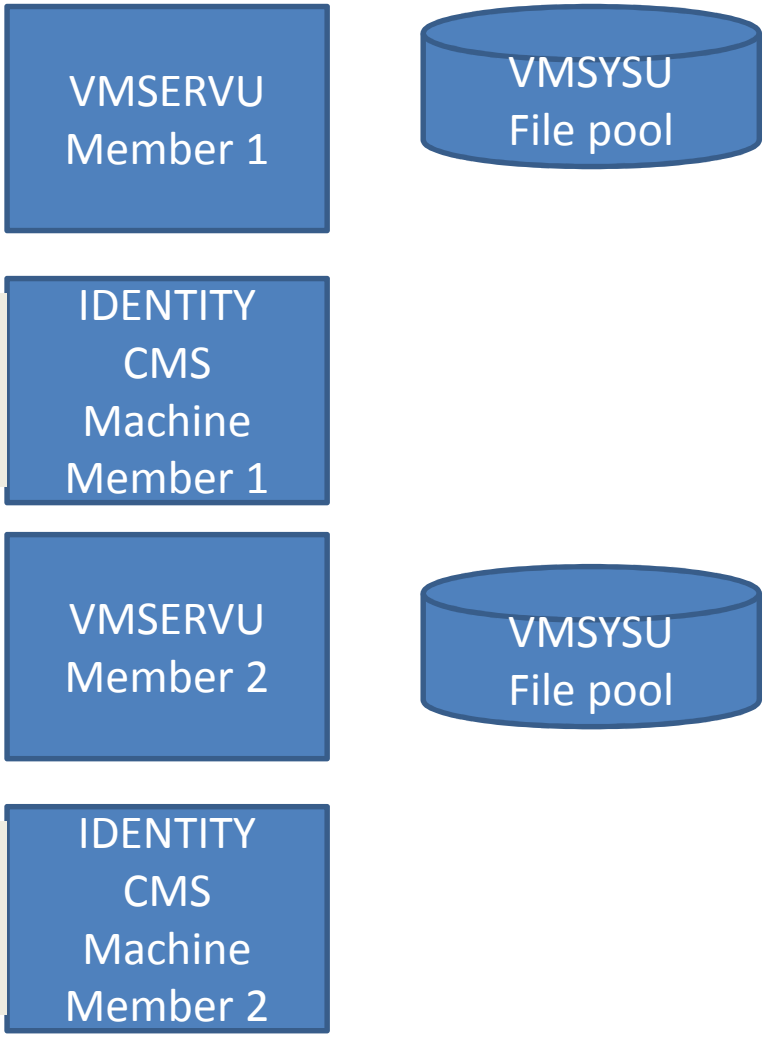


```
IDENTITY HOTRDR VMROCKS 64M 128M   G
INCLUDE IBMDFLT
IPL CMS PARM AUTOCR FILEPOOL VMSYSU
```

No MDISK statements needed. Space is allocated from the VMSYSU file pool (private on each system).

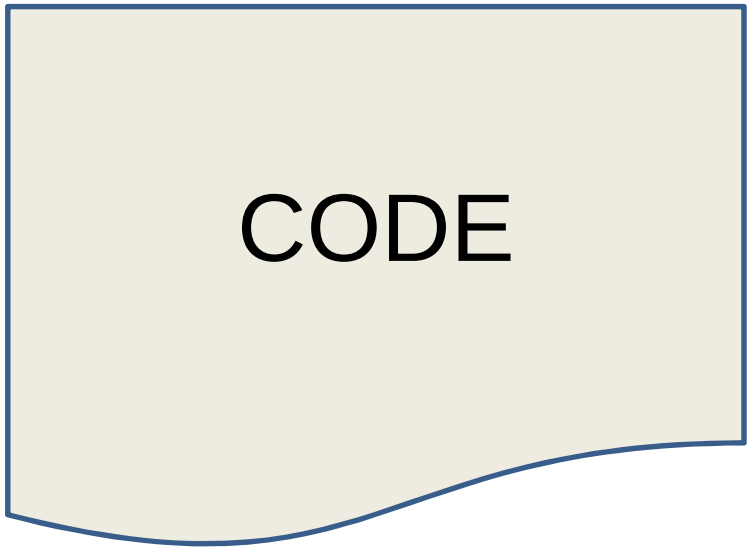
```
profile
HOTRDR   AT MEMBER1
Ready; T=0.01/0.01 16:40:51
```

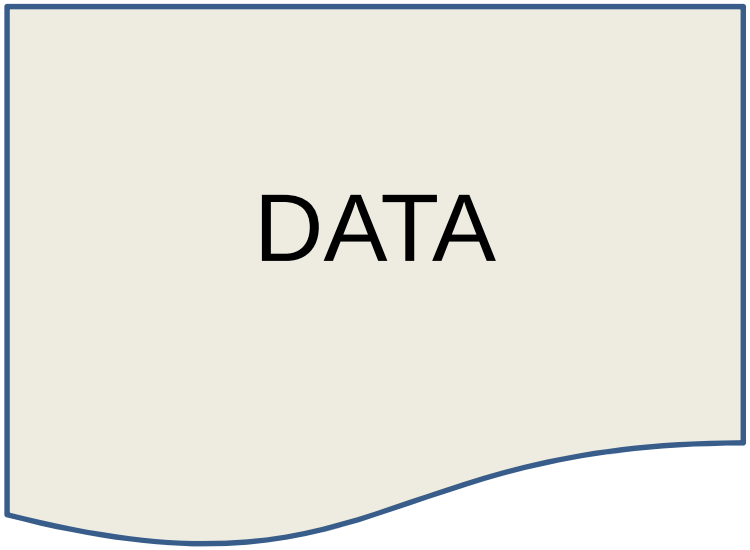
```
profile
HOTRDR   AT MEMBER2
Ready; T=0.01/0.01 16:40:51
```



Code is “read only” and references the NAMES file

- Keep variables externalized and generalized





VMRSRV01 NAMES file

- Keep variables externalized and generalized

```
:nick.DAVID
:list.david.kreuter@mythic.com

:nick.LPAR
:list.vmrtst01 vmrprd01 vmrprd02 vmrsrv01

:nick.DNDSUPRT
:list.joe.blow@mythic.com groupmailbox@mythic.com
david.kreuter@mythic.com
```

NAMES file referenced in the code

- Alerting exit in PERFSVM reads NAMES file

```
'pipe CP QUERY USERID ' ,
'|SPECS /NAMEFIND :nick 'listname ' ( file/ 1 w3 nw ',
'| CONSOLE',
'| COMMAND ',
'| LOCATE W1 /:list/',
'| CONSOLE',
'|SPECS /EXEC SENDFILE $ $ A TO/ 1 ',
' W2-* NW ',
" /(SMTP SUBJECT 'ALERT" sysid"'/ NW ",
'| CONSOLE',
'| COMMAND ',
'| CONSOLE'
```

```
>>> "pipe CP QUERY USERID |SPECS /NAMEFIND :nick dndsuprt (
file/ 1 w3 nw
| CONSOLE
| COMMAND
| LOCATE W1 /:list/
| CONSOLE
|SPECS /EXEC SENDFILE $ $ A TO/ 1
W2-* NW /(SMTP SUBJECT 'ALERT VMRSRV01'/ NW
| CONSOLE
| COMMAND
| CONSOLE"
```

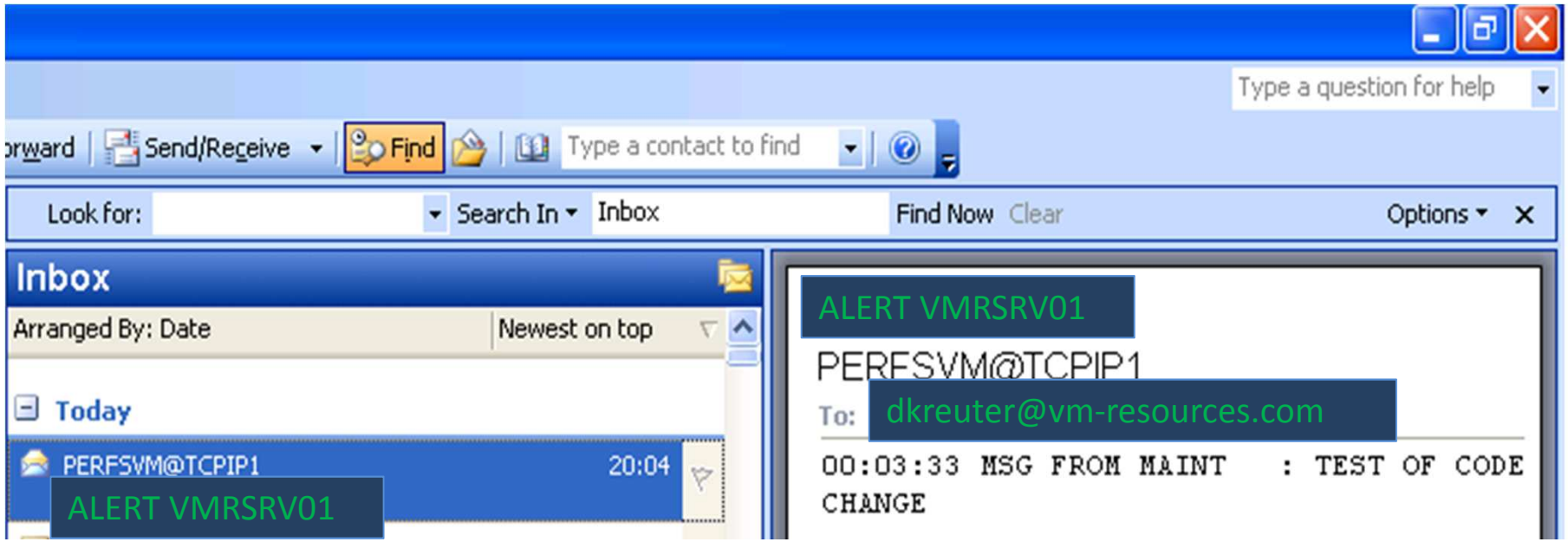
NAMEFIND :nick dndsuprt (file VMRSRV01

:list david.kreuter@mythic.com

EXEC SENDFILE $ $ A TO david.kreuter@mythic.com (SMTP SUBJECT 'ALERT VMRSRV01'

Variable substitution during execution

Result: The EMAIL in my INBOX



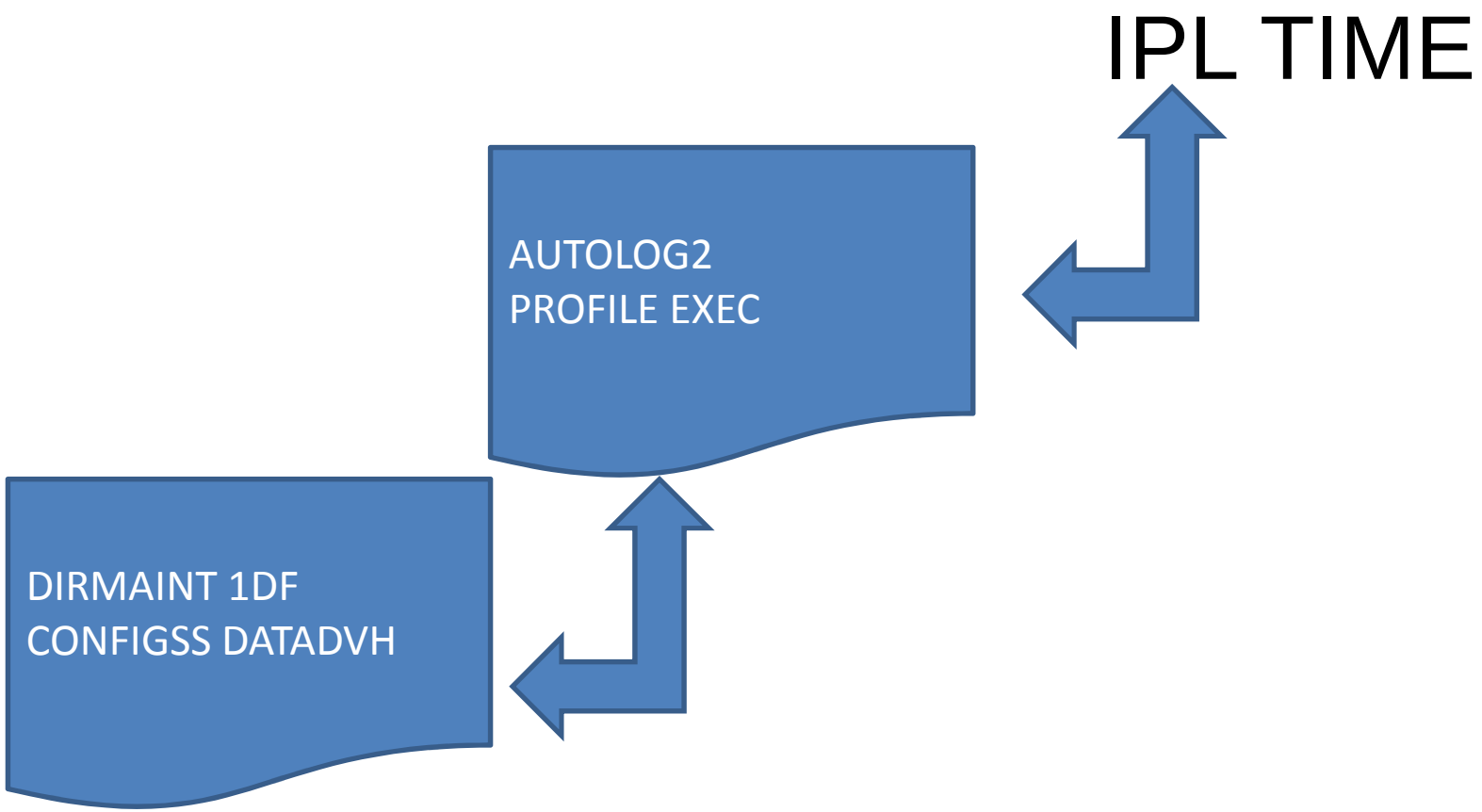
Alerts are sent from the ALERT EXEC on the PERFSVM 191 Minidisk.

DIRMAINT or DIRMSATx?

- Choice made for you at IPL time by AUTOLOG[12] !
- On DIRMSAT systems command is sent to member running DIRMAINT and response relayed back.

?

How the DIRMAINT Machine is Chosen



AUTOLOG2 PROFILE EXEC makes DIRMAINT or DIRMSAT decision

```
/* Call Dirm procedure to autolog the */
/* correct DIRMAINT/DIRMSAT */
Call Dirm

rrc = result

:

Dirm: Procedure
XAUTOLOG DIRMAINT
:
Call ssidirm
rrc = result
:
Ssirdirm:
….
```

IBM supplied source code. Will use the XAUTOLOG command to see if DIRMAINT is logged on in the cluster!

Where is DIRMAINT in Your Cluster?

```
at vmrsrv01 cmd q dirmaint
DIRMAINT - DSC
Ready; T=0.01/0.01 13:37:57
at vmrsrv02 cmd q n
DIRMAINT - SSI ,
:
PERFSVM  - DSC , RSCS    -   DSC , GCS       - DSC
TCPIP2   -  DSC , TCPIP1 -   DSC , FTPSERVE - DSC , TCPIP    - DSC
DIRMSAT2 - DSC , DTCVSW2 -  DSC , DTCVSW1   - DSC , VMSERVR  - DSC
VMSERVU  - DSC , VMSERVS -  DSC , RACFVM    - DSC , OPERSYMP - DSC
DISKACNT - DSC , OPERATOR - DSC
VSM - TCPIP
VSM - TCPIP2
Ready; T=0.01/0.01 13:38:08
```

DIRMAINT will be started on the first member to be IPLed. Other members will use a DIRMSAT machine. This can be changed by FORCing DIRMAINT and DIRMSAT machines and then starting DIRMAINT in a member of your choice followed by the DIRMSAT machines. Not recommended.

The CONFIGSS DATADVH

Used as DELTA to the base CONFIG DATADVH processed by DIRMAINT startup procedures.

```
* CONFIGSS DATADVH
* Created at install time to define a Satellite Server
* for each member
*
*********************************************************************

SATELLITE_SERVER= DIRMSAT     VMRSRV01
SATELLITE_SERVER= DIRMSAT2    VMRSRV02
DATAMOVE_MACHINE= DATAMOVE    VMRSRV01 *
DATAMOVE_MACHINE= DATAMOV2    VMRSRV02 *
```

From the DIRMSAT System

```
dirm for maint review
DVHXMT1191I Your REVIEW request has been sent for
processing to DIRMAINT
DVHXMT1191I at VMRSRV01 via DIRMSAT2.
Ready; T=0.01/0.01 13:43:00
DVHREQ2288I Your REVIEW request for MAINT at *
DVHREQ2288I has been accepted.
DVHREQ2289I Your REVIEW request for MAINT at *
DVHREQ2289I has completed; with RC = 0.
RDR FILE 0443 SENT FROM DIRMAINT PUN WAS 0034 RECS 0474
CPY 001 A NOHOLD NOKEEP
```

The DIRM request is handled by the DIRMSAT2 machine. DIRMSAT2 relays the command to the member with DIRMAINT via spool.

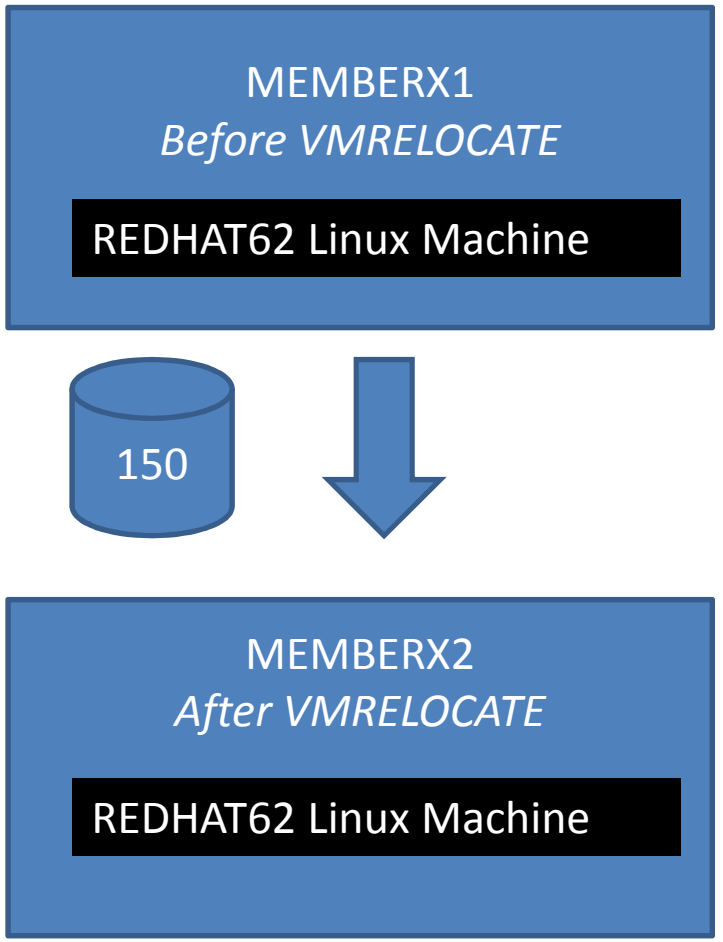
Relocation Observed

- Once the technical criterion are met you are good to go!
- Used extensively for maintenance schedules.
  - Not so much for forced load balancing.
- Interfaces have ease of use.
  - Must deal with EQIDs.
  - DETACH non-Linux MDISKS

VMRELOCATE

The Payoff: Virtual Machine Relocation

- Live Guest Relocation is glorious
- Many conditions must be met on source and target cluster members:
  - Sufficient memory and paging bandwidth on target
  - EQID on network devices (Equivalency IDs)
  - CMS minidisks are a nuisance mostly, detach before the move.
- Define same VSWITCH name on all members
- Seamless move, applications stay alive!



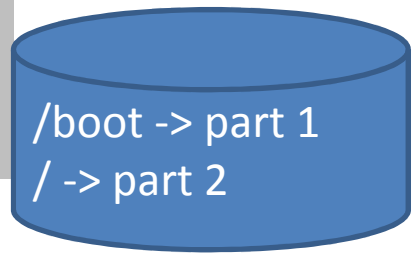
The REDHAT62 Directory Entry

- Single Configuration Userid
- 150 Disk with the root filesystem

```
USER REDHAT62 VMRULES 512M 12G G
   IPL 150 LOADPARM 0
   MACHINE XA
   OPTION CHPIDV ONE
   CONSOLE 0009 3215 T DAVE
   NICDEF 0600 TYPE QDIO LAN SYSTEM CLUSTNET
   SPOOL 000C 2540 READER *
   SPOOL 000D 2540 PUNCH A
   SPOOL 000E 1403 A
   MDISK 0150 3390 0 4999 0X0150 MR REDHAT62
```

Simulate CHPID and DASD path virtualization

VSWITCH



MEMBERX1 REDHAT62 Relocation test

```
vmrelocate test user redhat62 to memberx2
User REDHAT62 is eligible for relocation to MEMBERX2
Ready; T=0.01/0.01 14:46:44


q vswitch
VSWITCH SYSTEM CLUSTNET Type: QDIO    Connected:
  VLAN Unaware
  MAC address: 02-00-01-00-00-05     MAC Protection: OFF
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF
 Uplink Port:
  RDEV: E000.P00 VDEV: 0600 Controller: DTCVSW2
        EQID: ABCDEFGH
```

Matches names on VSWITCHes on other MEMBERs

Unique MACID for VSWITCH

Matches names on OSAs on other members

MEMBERX1 REDHAT62

```
q vswitch user redhat62
;
Adapter Connections:
    Adapter Owner: REDHAT62 NIC: 0600.P00 Name: UNASSIGNED  Type:
QDIO
      RX Packets: 19         Discarded: 0          Errors: 0
      TX Packets: 8          Discarded: 0          Errors: 6
      RX Bytes: 3815                TX Bytes: 636
      Device: 0602  Unit: 002   Role: DATA       Port: 0001
      Options: Broadcast Multicast IPv6 IPv4 VLAN
        Unicast IP Addresses:
          10.100.0.103  IP      MAC: 02-00-01-00-00-06  MACID
          FE80::200:100:100:6  MAC: 02-00-01-00-00-06 Local
        Multicast IP Addresses:
          224.0.0.1            MAC: 01-00-5E-00-00-01
          FF02::1              MAC: 33-33-00-00-00-01 Local
          FF02::1:FF00:6       MAC: 33-33-FF-00-00-06 Local
```

MEMBERX2 VSWITCH before relocation of REDHAT62

```
q vswitch
VSWITCH SYSTEM CLUSTNET Type: QDIO    Connected: 0    Maxconn:
INFINITE
  PERSISTENT  RESTRICTED    NONROUTER            Accounting: OFF
  USERBASED
  VLAN Unaware
  MAC address: 02-00-02-00-00-03    MAC Protection: OFF
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF
 Uplink Port:
  RDEV: E000.P00 VDEV: 0600 Controller: DTCVSW1
        EQID: ABCDEFGH
```

Matches names on VSWITCHes on other MEMBERs

Unique MACID for VSWITCH

Matches names on OSAs on other members

Ka-Ching: The Relocate Pays Off!

```
14:51:09 vmrelocate move user redhat62 to memberx2
14:51:09 Relocation of REDHAT62 from MEMBERX1 to MEMBERX2
started
14:51:11 User REDHAT62 has been relocated from MEMBERX1
to MEMBERX2
14:51:11 Ready; T=0.01/0.01 14:51:11
```

Notice that it took 2 Seconds

send cp redhat62 q userid

REDHAT62: REDHAT62 AT MEMBERX2

```
q vswitch user redhat62
VSWITCH SYSTEM CLUSTNET Type: QDIO    Connected: 1    Maxconn: INFINITE
  :
  MAC address: 02-00-02-00-00-03    MAC Protection: OFF
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF
 Uplink Port:
  RDEV: E000.P00 VDEV: 0600 Controller: DTCVSW1
        EQID: ABCDEFGH
 Adapter Connections:
    Adapter Owner: REDHAT62 NIC: 0600.P00 Name: UNASSIGNED  Type: QDIO
:
      Device: 0602  Unit: 002   Role: DATA       Port: 0001
      Options: Broadcast Multicast IPv6 IPv4 VLAN
        Unicast IP Addresses:
    IP  10.100.0.103          MAC: 02-00-01-00-00-06  MACID
        FE80::200:100:100:6   MAC: 02-00-01-00-00-06 Local
```

Matches names on OSAs on other members

Some Lessons Learned

- IDENTITY machines and SFS
- Mapping minidisks
- DIRMAINT usage
- RACF usage
- VMRELOCATE ease of use
- Lots of work managing multiple LPARs!

Today’s Presentation As Advertised

- By the time we're gathered in the heat of the Workshop many of us will be running z/VM 620. In this presentation David will discuss the usage of z/VM 620 at his clients. As usual David's technical drill down will be replete with information on clustering system design, LPAR setup, tool smithing, networking, and some slick methods of dealing with multiple clusters.

Thanks and Kudos

- Dave Jones for standing up and delivering in David Kreuter's absence!
- Len Santalucia for his support and sponsorship!
- David Kreuter says "sorry I missed you and see you next year!"

For More Information please contact…

Len Santalucia, CTO & Business Development Manager
Vicom Infinity, Inc.
One Penn Plaza – Suite 2010
New York, NY 10119
212-799-9375 office
917-856-4493 mobile
lsantalucia@vicominfinity.com

About Vicom Infinity

Account Presence Since Late 1990's
IBM Premier Business Partner
Reseller of IBM Hardware, Software, and Maintenance
Vendor Source for the Last 4 Generations of Mainframes/IBM Storage
Professional and IT Architectural Services
Vicom Family of Companies Also Offer Leasing & Financing, Computer Services, and IT Staffing & IT Project Management