

Non-disruptively Migrating Linux Guests in Their Entirety

Michael MacIsaac
INNOVATION Data Processing
mmacisaac@fdrinnovation.com

Metropolitan VM Users Association
Summer Meeting, 15 July 2014
Marist College, Poughkeepsie, NY



© Copyright 2014 INNOVATION Data Processing. All rights reserved.



Agenda

- **Introductions**
- Hierarchy of availability
- Business continuance tools
- Platform convergence
- Estimating migration effort
- Summary



Introductions



- Who am I?
 - Michael MacIsaac
 - Product Manager for z/VM and Linux
 - mmacisaac@fdrinnovation.com
- Who are you?
 - An Innovation Data Processing customer?
 - An FDRPAS on z/OS customer?
 - Have z/VM & Linux in production/test/PoC?
 - A z/VM & Linux only shop?

Hierarchy of Availability

- Hierarchy of availability (lower to higher)
 - High Availability
 - Continuous Operations
 - Continuous Availability



Source: "High Availability Architectures For Linux on IBM System z" Version 2, June 15, 2010 by Steve Wehr, Scott Loveland and Harriet Morrill of IBM

© Copyright 2014 INNOVATION Data Processing. All rights reserved.



High Availability



- High Availability (HA)
 - Provides service during defined periods, at agreed upon levels (SLAs)
 - Recovery Time Objective (RTO)
 - Recovery Point Objective (RPO)
 - Avoids **unplanned outages**
 - Employs failure detection, automatic recovery/failover, problem/change management, etc.

Continuous Operations



- Continuous Operations (CO)
 - Avoids **planned outages**
 - Employs non-disruptive hardware and software upgrades and configuration changes

Continuous Availability



- Continuous Availability (CA)
 - Delivers non-disruptive service to the end user, 24 hrs/day x 365 days/yr
 - No ***planned*** nor ***unplanned*** outages
 - Continuous operations + redundancy of any single point of failure and failover to the redundant components



Tools in Your HA Toolbox

- Resilient hardware with dynamic features
 - Mainframe, PR/SM, standby memory/CPU, etc.
- Disk local mirroring and remote replication tools
- Resiliency z/VM and Linux features
 - Hot plugging memory, CPU, file systems
- HA software
 - Oracle RAC, IBM WAS XD, IBM DB2 HADR, etc.
- Business continuance tools
 - z/VM 6.2+ SSI and LGR
 - Innovation FDRPAS for z/OS & FDRPASVM for z/VM

Agenda

- Introductions
- Hierarchy of availability
- **Business continuance tools**
 - z/VM SSI and LGR
 - **FDRPASVM non-disruptive migration**
- Platform convergence
- Estimating migration effort
- Summary

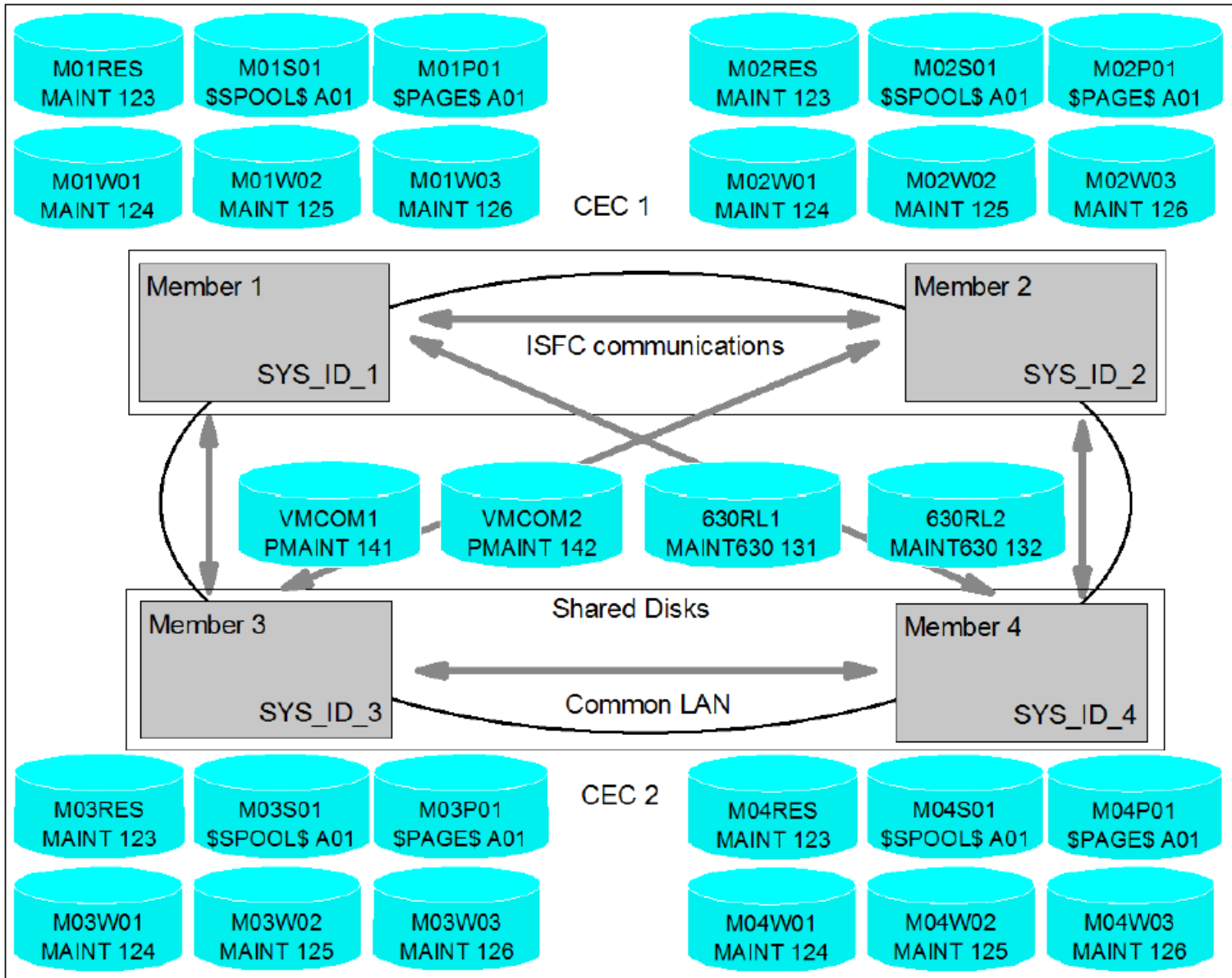


z/VM SSI and LGR



- Single System Image (SSI)
 - 2-4 z/VM “member” systems share and coordinate resources
 - This becomes an “SSI cluster”
- Live Guest Relocation (LGR)
 - Running Linux systems can move cross-LPAR or CEC
 - Memory and CPU are moved, but not disk
- Can reduce or prevent planned outages

z/VM SSI block diagram



FDRPASVM



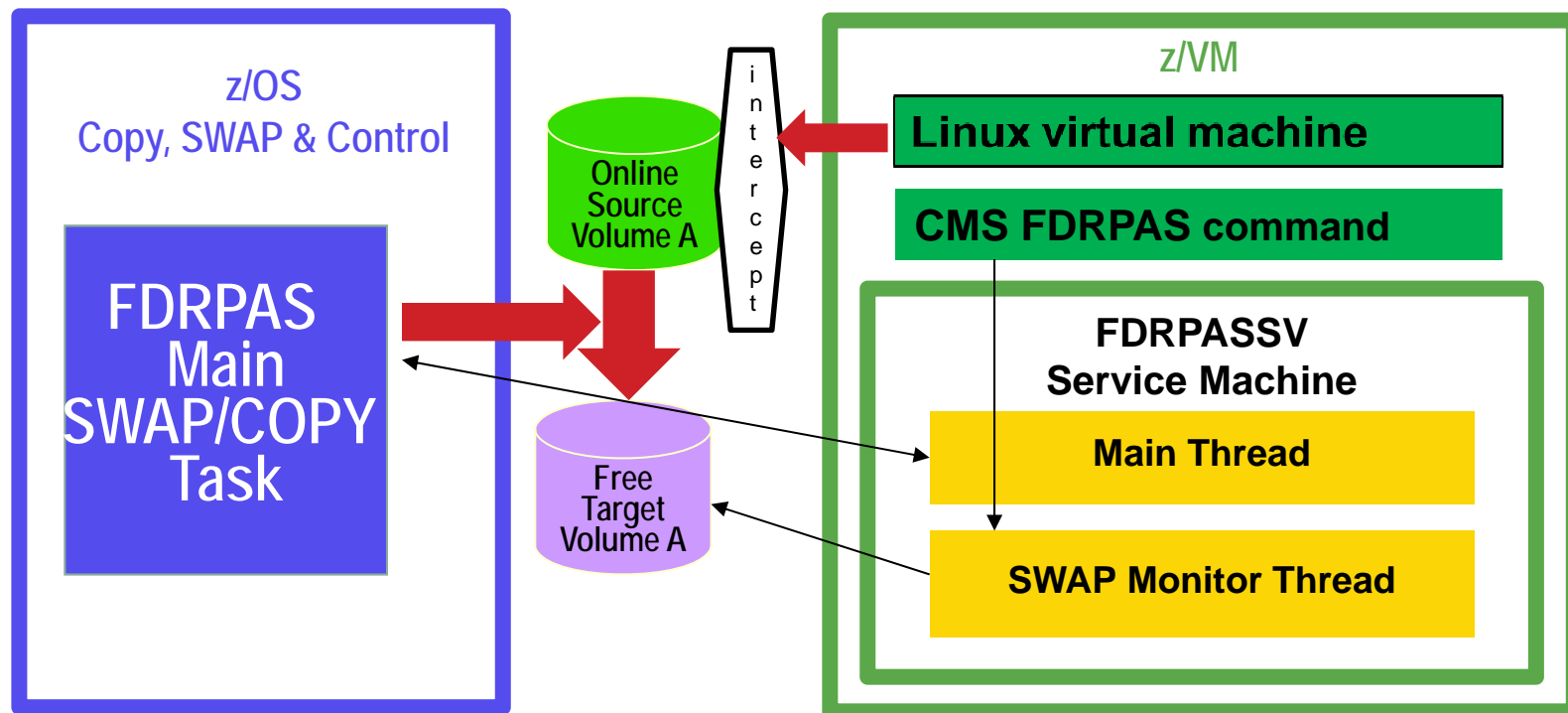
- Non-disruptively migrate DASD of running systems
 - Copies entire source volume(s) to target(s)
 - Monitors changed tracks on source volume
 - Copies changed tracks
 - Swaps all I/O operations to use target volume(s)
- Beta tested at multiple sites in 2013
- GA in January 2014
- Supports z/VM 5.4, 6.2 and 6.3
- Move to a new DASD storage unit non-disruptively

FDRPASVM functions



- FDRPASVM supports migration of
 - Minidisk volumes (PERM)
 - Full-pack and DEDICATED volumes
 - Smaller to larger volumes (ex: 3390-9 to 3390-27)
- FDRPAS functions
 - SIMSWAP – Simulate and validate copy and swap
 - SIMSWAPMON – Simulate and validate monitoring updates
 - SWAPDUMP – Create point-in-time copy of volume(s)
 - SWAP – Copy and swap volume(s)

FDRPASVM components

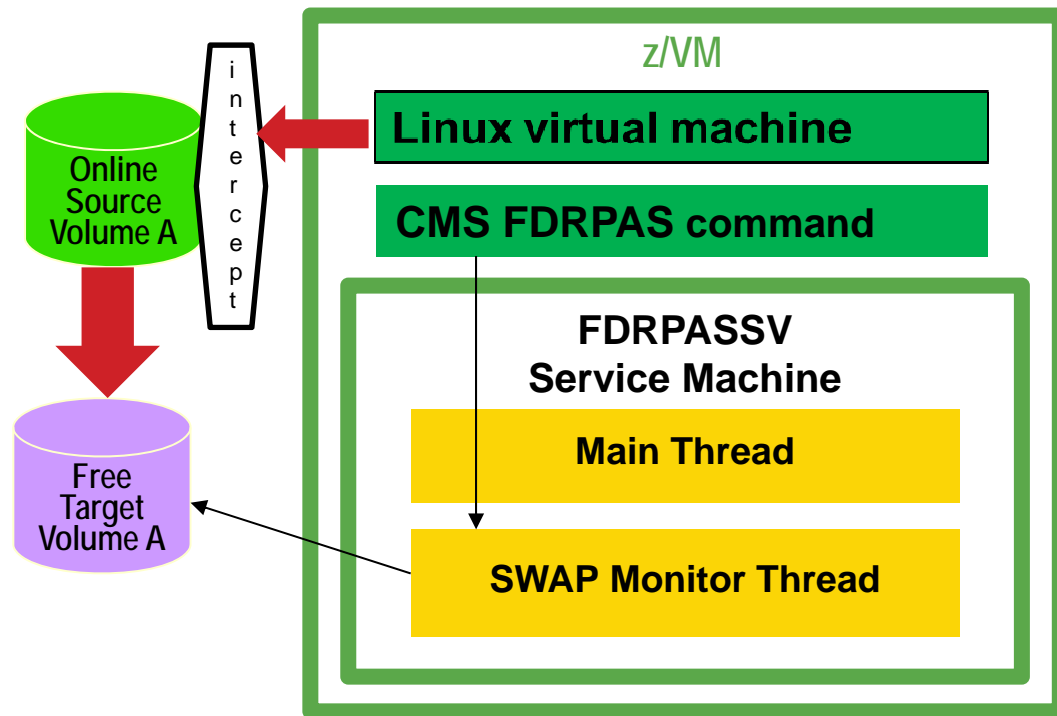


FDRPASVM component detail



- z/VM detail
 - Tracks changes to source device and swaps volser
 - System requirements
 - Service machine (FDRPASSV) is running
 - Source volume is **online**
 - Target volume is **online** and **FREE**
 - Monitor program started with FDRPAS command
- Start monitor on LPARs w/access to target volumes
- z/OS detail
 - Copies to target and recopies changed tracks
 - System requirements
 - Same source volume is **online**
 - Same target volume is **offline**
 - Invoke **FDRPAS** job using JCL or ISPF

FDRPASVM Block Diagram





FDRPASVM setup

- Service machine (FDRPASSV) is running
 - Logon to FDRPASSV interactively

```
...  
DIAGNOSE 104 ALREADY DEFINED  
...  
PASIUCSM020I WAITING FOR AN EVENT TO PROCESS
```

- Start FDRPASSV on AUTOLOG1 191 disk (mode F)

```
==> x profile exec f  
...  
/*****/  
/* Customer processing can be added here */  
/*****/  
"CP XAUTOLOG TCPIP" /* Autolog TCPIP */  
"CP SET SIGNAL SHUTDOWN 300" /* Allow guests 5 min to shut down */  
"CP XAUTOLOG FDRPASSV" /* Start the FDRPASSV service machine */
```



FDRPASVM on z/VM volumes

- Example of swapping Linux on rdev **1887** to **B887**
- Source volume is **online**
- Target volume is online and **FREE**
 - Use CP QUERY <rdev> and DETACH commands:

```
==> q 1887 b887
DASD 1887 CP SYSTEM VM1887 2
DASD B887 CP SYSTEM VMB887 0
==> detach b887 system
DASD B887 DETACHED SYSTEM
==> q 1887 b887
DASD 1887 CP SYSTEM VM1887 2
DASD B887 VMB887
```

FDRPASVM on z/VM monitoring



- Monitor target volume (e.g. from MAINT)

- Access FDRPAS CMS command:

```
...  
'EXEC VMLINK PASMAINT 691'  
'SET LANG (ADD PAS USER'  
...
```

- Issue FDRPAS command for target volume:

```
==> fdrpas monitor type swap b887
```

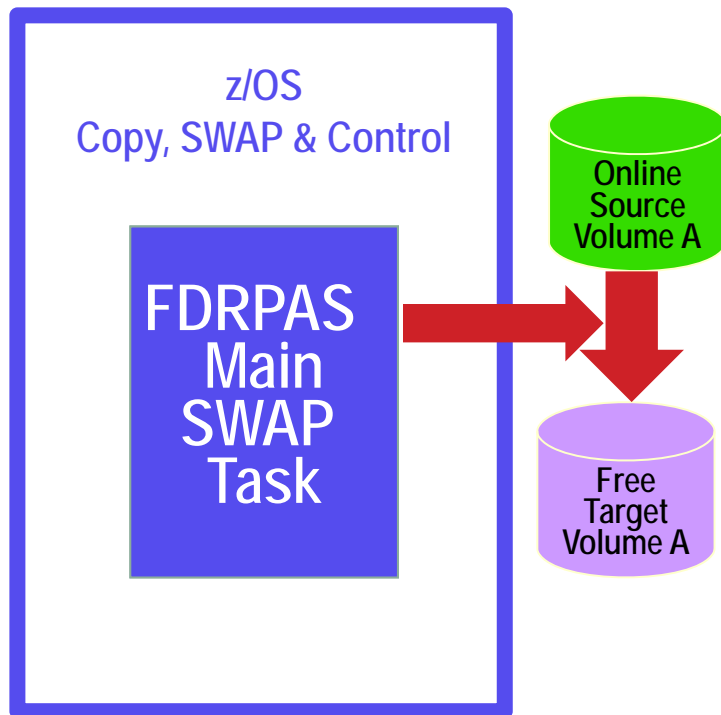
```
...  
REQUEST ACCEPTED  
SEVERING IUCV CONNECTION  
...
```

```
* MSG FROM FDRPASSV: PASIUCSM009I 1 ELIGIBLE DEVICE(S) FOUND
```

- Watch console on FDRPASSV

```
...  
PASMONVW080I DEVICE B887(B887) WAITING FOR SWAP INITIATION
```

FDRPASVM z/OS components



FDRPAS z/OS view of z/VM volumes



- Source volume should be **online**
- Target volume is **offline**
 - Use DISPLAY and VARY commands

```
==> d u,,,1887
```

UNIT	TYPE	STATUS	VOLSER	VOLSTATE
1887	3390	OFFLINE		/RSDNT

```
==> d u,,,B887
```

UNIT	TYPE	STATUS	VOLSER	VOLSTATE
B887	3390	OFFLINE		/RSDNT

```
==> v 1887,online
```

```
IEE302I 1887      ONLINE
```

```
==> d u,,,1887
```

UNIT	TYPE	STATUS	VOLSER	VOLSTATE
1887	3390	O	VM1887	PRIV/RSDNT

FDRPAS on z/OS startup



- Invoke FDRPAS job in one of two ways
 - From a JCL job

```
==> submit
//PASTEST1 JOB ('PR=YES'), 'ME', CLASS=M,
//  NOTIFY=ME
//*
//*****
//*   FDRPAS                                          *
//*****
//*
//PASB      EXEC  PASPROC
//PAS.SYSIN DD  *
*SIMSWAP  TYPE=FULL, LARGERSIZE=OK, MAXTASKS=4, NONRESPONDING=FAIL
*SIMSWAPMON TYPE=FULL, LARGERSIZE=OK, MAXTASKS=4, NONRESPONDING=FAIL
*SWAPDUMP TYPE=FULL, LARGERSIZE=OK, MAXTASKS=32, NONRESPONDING=FAIL
  SWAP  TYPE=FULL, LARGERSIZE=OK, MAXTASKS=32, NONRESPONDING=FAIL
  MOUNT VOL=VM1887, SWAPUNIT=B887
```

FDRPASVM Process Review



- FDRPAS and FDRPASVM “plumbing”
 - z/VM “intercepts” installed to monitor source volume changes
 - z/OS main SWAP task copies source to target volume
 - FDRPASSV swap thread passes changes to z/OS main SWAP
 - z/OS main SWAP task recopies changed tracks
 - z/VM HYPERSWAP is issued when source and target are in sync
 - Target volume becomes the source volume transparently
 - FDRPASSV intercepts are removed

FDRPASVM z/VM Output



- MAINT on z/VM

- Messages from FDRPASSV:

...

* MSG FROM FDRPASSV: PASMONVT233I VMLAB63B (SERIAL# 04E2062818)
ACKNOWLEDGES THE SWAP OF VOL=VM1887 AND HAS JOINED IN SWAP OF UNIT=1887
TO B887

* MSG FROM FDRPASSV: PASMONVT241I FDRPAS SUCCESSFULLY COMPLETED SWAP OF
VOL=VM1887 TO UNIT=B887

- Query source and target devices again:

==> **q 1887 B887**

DASD 1887 **FDR3VM**

DASD B887 CP SYSTEM **VM1887** 2

FDRPAS z/OS Output



- JCL output (syslog)

```
FDR233 CPUB      (SERIAL# 02E2062818) ACKNOWLEDGES THE SWAP OF VOL=VM1887  - HTC
2107900 TO HTC 2107900
FDR233 VMLAB63B (SERIAL# 04E2062818) ACKNOWLEDGES THE SWAP OF VOL=VM1887 AND HA
S JOINED IN SWAP OF UNIT=1887 TO B887
...
OPERATION STATISTICS FOR 3390 VOLUME.....VM1887
      CYLINDERS ON VOLUME.....10,017
      DATASETS PROCESSED.....0
      BYTES READ FROM DASD.....7,593,410,036
      DASD TRACKS SWAPPED.....154,127
      UPDATED TRACKS RECOPIED.....3,873
      DASD EXCPS.....10,418
      TARGET DASD EXCPS.....10,371
      CPU TIME (SECONDS).....2.257
      ELAPSED TIME (MINUTES).....2.6
      SWAP TIME.....2.4

FDR SUCCESSFULLY COMPLETED
```

FDRPASVM trial

- To request a free trial:
 - <http://www.fdr.com/riskfreetrial/index.cfm>



Agenda

- Introductions
- Hierarchy of availability
- Business continuance tools
- **Platform convergence**
- Estimating migration effort
- Summary



Automating z/OS job submission from z/VM



- To drive z/OS JCL jobs from z/VM
- New paper describing how to:
 - Submit JCL job(s) from z/VM to z/OS
 - Using FILETYPE=JES mode of the z/OS FTP server
 - Using the VMFTP tool to process output from FTP
 - All in a single REXX “wrapper”
 - Second wrapper for multiple job submissions
- See http://www.fdr.com/Manuals_CurrentVersion/JCLfromVM.pdf

FTP session overview

- Submit JCL jobs through FTP

```
==> ftp zOS.ftp.server
ftp> z/OS credentials
...
ftp> site filetype=jes
...
ftp> put myjob.jcl
...
ftp> get jobid.x
...
ftp> quit
```



FTPJOB EXEC



- **Wrap JCL job submission in a REXX EXEC**

```
/* EXEC to submit a JCL job using the VMFTP tool          */
Parse upper arg jobName .
If (jobName = '') Then Do                                /* no job name passed in */
  Say 'Error: expected parameter JobName not found'
  Return 1
End
'STATE' jobName 'JCL *'                                 /* check that file exists */
if (rc <> 0) Then Do                                    /* file not found => exit */
  Say 'Error: File' jobName 'JCL * not found'
  Return 2
End
'VMFTP FTPJOB (PARM' jobName                          /* Invoke the VMFTP macro */
```

FTPJOB VMFTP

- Use VMFTP environment

```
/* VMFTP Macro to submit a JCL job and extract output */
Parse upper arg jobName . /* get the one argument */
system = 'myzos' /* target z/OS system */
userID = 'myuserid' /* z/OS user ID */
password = 'mypasswd' /* password: case sensitive */
jobFile = jobName||'.JCL' /* input file */
/* do the work */
/* start the FTP session */

'open' system /* send the user ID */
...
userID /* send the password */
...
password /* set server to JCL mode */
...
'site filetype=jes'
```



FTPJOB VMFTP (cont'd)

```
'put' jobFile
```

```
...
```

```
jobNumber = Word(output.4, 7)
```

```
...
```

```
'get' jobNumber||'.X'
```

```
...
```

```
'quit'
```

```
...
```

```
/* send the JCL job */
```

```
/* get job # from output */
```

```
/* retrieve the job output */
```

```
/* end the FTP session */
```



A sample z/OS JCL job



```
//MDMS JOB ('PR=YES'), 'MIKE', CLASS=M, MSGCLASS=X
//STEP1 EXEC PGM=SORT
//SYSIN DD *
  SORT FIELDS=(1,75,CH,A)
/*
//SYSOUT DD SYSOUT=*
//SORTIN DD *
MERCURY
VENUS
EARTH
MARS
JUPITER
SATURN
URANUS
NEPTUNE
/*
//SORTOUT DD SYSOUT=*
/*
```



Example of submitting one JCL job

```
==> ftpjob mdms
```

```
Job MDMS has been submitted - waiting for output ...
```

```
Output saved to JOB04838 X A
```

```
==> type job04838 x a
```

```
...
```

```
!! END OF JES SPOOL FILE !!
```

```
EARTH
```

```
JUPITER
```

```
MARS
```

```
MERCURY
```

```
NEPTUNE
```

```
SATURN
```

```
URANUS
```

```
VENUS
```

```
!! END OF JES SPOOL FILE !!
```


Submitting multiple JCL jobs

- Put a wrapper around FTPJOB
- Input file:

```
==> type ftpjobs list
```

```
mdms
```

```
mdmc
```

```
nosuch
```



FTPJOBS EXEC

```
/* EXEC to automate JCL job submission using FTP and VMFTP tool */
jobsFile = 'FTPJOBS LIST A' /* file with JCL jobs to run */
...
Do i = 1 To jobs.0 /* check line for error msg */
  Say 'Submitting job' jobs.i /* info message */
  'STATE' jobs.i 'jcl' /* check if file exists */
  If (rc = 0) Then /* file exists */
    Call FTPjob jobs.i /* FTPjob submits job */
  End
Exit
FTPjob: procedure /* to submit one job */
Parse arg jobName /* get job name */
'VMFTP FTPJOB (PARM' jobName /* use VMFTP macro */
...
Return
```





Example of submitting multiple JCL jobs

```
==> ftpjobs
```

```
Submitting job mdms
```

```
Job MDMS has been submitted - waiting for output ...
```

```
Output saved to JOB04841 X A
```

```
Submitting job mdmc
```

```
Job MDMC has been submitted - waiting for output ...
```

```
Error messages in JOB04842 X A:
```

```
    FDR316**    FDR  DID NOT FIND REQUESTED MOUNT SWAP VOL=73BSP1
```

```
...
```

```
Submitting job nosuch
```

```
DMSSTT002E File NOSUCH JCL not found
```

Agenda

- Introductions
- Hierarchy of availability
- Business continuance tools
- Platform convergence
- **Estimating migration effort**
- Summary





Estimating migration effort

- How much DASD space are on my systems?
- CALCDASD EXEC identifies type and size of DASD
 - Needs no arguments if all DASD “belongs” to z/VM
 - ==> `calcdasd`
 - Can take rdev-range if not all DASD “belongs” to z/VM
 - ==> `calcdasd 1880-1887`
- Counts 3390-1s, -2s, -3s, -9s –As (EAVs) and “other sizes”
- Identifies CP-Owned, SYSTEM and ATTACHED disks
- Summarizes findings



3390 DASD sizes

Name	Cylinders	Notes
3390-1	1113	
3390-2	2226	
3390-3	3339	
3390-9	10017	
Small 3390-27	30051	3x size of 3390-9
3390-27	32760	Aka 3390-32k, not multiple of 1113
Small 3390-54	60102	6x size of 3390-9
3390-54	65520	Aka 3390-64k, largest directory size
Non-standard	< 65520	Any other size smaller than 65520
3390-A	> 65520	Any other size larger than 65520



CALCDASD help

==> **calcdasd help**

Synopsis:

Calculate the amount and types of DASD on a system and report

Syntax is:

```
>>--CALCDASD--+-----+--+-----+--><
      | -rdev----- |      |
      | <-----< |      +-- (---+---HELP---+-'
      '-rdev1-rdev2-----'      '---SHORT---'
```

Where:

- no rdev range: Query all DASD
- rdev range: Query these device addresses, not all DASD
- SHORT option: Do not include QUERY DASD output
- HELP option: This screen

CALCDASD - short output

```
==> calcdasd (short
```

```
Number of DASD models:
```

```
3390-1s      (1113 cylinders):  0
3390-2s      (2226 cylinders):  0
3390-3s      (3339 cylinders):  0
3390-9s      (10017 cylinders):  8
3390-As      (sizes > 65520):  0
```

```
...
```





CALCDASD - short output (cont'd)

CP-owned cylinder allocation:

Type	Cylinders	GB	% used
----	-----	----	-----
TDISK	0	0.00	0.00
PAGE	10016	8.51	8.49
SPOOL	10016	8.51	1.84
DRCT	20	0.02	20.00

Total cylinder allocation:

Type	Cylinders	GB
----	-----	----
CP-OWNED	40068	34.06
SYSTEM	40068	34.06
ATTACHED	0	0.00
	-----	-----
Total:	80136	68.11

CALCDASD – rdev range

==> **calcdasd 1880-1887**

QUERY 1880-1887 output:

```
DASD 1880 CP OWNED 63BRES
DASD 1881 CP OWNED 63BCOM
DASD 1882 CP SYSTEM 63BREL
DASD 1883 CP OWNED 63BSP1
DASD 1884 CP OWNED 63BPG1
DASD 1885 CP SYSTEM 63BW01
DASD 1886 CP SYSTEM VM1886
DASD 1887 CP SYSTEM VM1887
```

Number of DASD models:

```
3390-1s (1113 cylinders): 0
3390-2s (2226 cylinders): 0
3390-3s (3339 cylinders): 0
3390-9s (10017 cylinders): 8
3390-As (sizes > 65520): 0
```

...





CALCDASD – rdev range (cont'd)

CP-owned cylinder allocation:

Type	Cylinders	GB	% used
----	-----	----	-----
TDISK	0	0.00	0.00
PAGE	10016	8.51	8.49
SPOOL	10016	8.51	1.84
DRCT	20	0.02	20.00

Total cylinder allocation:

Type	Cylinders	GB
----	-----	----
CP-OWNED	40068	34.06
SYSTEM	40068	34.06
ATTACHED	0	0.00
-----		-----
Total:	80136	68.11

...



CALCDASD – different sizes

==> **calcdasd**

...

Warning - non standard size: 048E has 556 cylinders
Warning - non standard size: 048F has 556 cylinders
Warning - non standard size: 8F10 has 20034 cylinders
Warning - non standard size: 8F11 has 20034 cylinders
Warning - non standard size: 8F12 has 20034 cylinders
Warning - non standard size: 8F13 has 20034 cylinders
Warning - non standard size: 8F14 has 20034 cylinders

Number of DASD models:

3390-1s	(1113 cylinders):	2
3390-2s	(2226 cylinders):	2
3390-3s	(3339 cylinders):	11
3390-9s	(10017 cylinders):	124
3390-27s	(32760 cylinders):	4
3390-54s	(65520 cylinders):	24
non-std	(sizes < 65520):	7
3390-As	(sizes > 65520):	0

CALCDASD EXEC

- To get a copy, email:
 - support@fdrinnovation.com



Summary

- User testimonial
- Benefits
- Resources
- Q & A



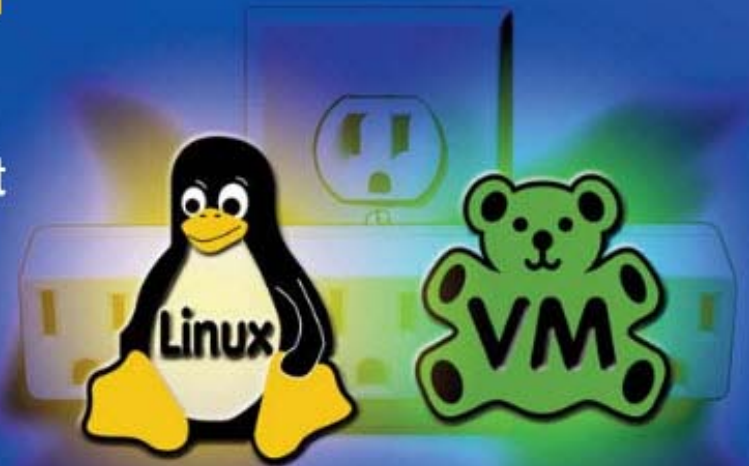
User Testimonial



Non-Disruptive Migration

“The business units requirements that rely on Linux volumes have made it very difficult for us to schedule outages to move their systems. FDRPASVM now allows us to move them non disruptively like FDRPAS does for our z/OS volumes.”

A Large Financial Company



Benefits of FDRPASVM



- FDRPAS technology has a proven record of reliability
- Used in 1700+ data centers since 2001
- Supports concurrent processing:
 - Of many volumes
 - By many users
- You don't have to bring z/VM* or Linux systems down

FDRPASVM Futures



- Swapping smaller to larger volumes:
 - Volume allocation table to reflect all PERM space (2Q 14)
 - Today: reflects source volume allocation table
- CP-Owned volumes:
 - To be supported (2Q 14)

Resources

- Manuals
 - FDRPASVM V5.4L80 User Manual
<http://www.fdr.com/FDRPASVMdoc.pdf>
 - FDRPAS, FDRMOVE, and FDRERASE Manual
http://www.fdr.com/Manuals_CurrentVersion/FDRPAS_V54L80.pdf
- This presentation
<http://www2.marist.edu/~mvmua/14071515.pdf>
- FDR demos
<http://www.innovationdp.fdr.com/index.cfm?hptab=4#>
Click **View the FDRPAS product demo**
- Risk-free Trial
http://www.innovationdp.fdr.com/riskfreetrial/form_rft.cfm
Choose “FDRPASVM product”
- My e-mail address
mmacisaac@fdrinnovation.com



Thank You



CORPORATE HEADQUARTERS: 275 Paterson Ave., Little Falls, NJ 07424 • (973) 890-7300 • Fax: (973) 890-7147
E-mail: support@fdrinnovation.com • sales@fdrinnovation.com • <http://www.innovationdp.fdr.com>

EUROPEAN OFFICES:	FRANCE 01-49-69-94-02	GERMANY 089-489-0210	NETHERLANDS 036-534-1660	UNITED KINGDOM 0208-905-1266	NORDIC COUNTRIES +31-36-534-1660
--------------------------	--------------------------	-------------------------	-----------------------------	---------------------------------	-------------------------------------