#SHAREorg

SHARE
Technology · Connections · Results

Managing SAN for Linux on z/VM
A Nationwide Perspective

James Vincent
Nationwide Insurance

March 2012

Disclaimer and C.Y.A.

The content of this presentation is for information only and is not intended to be an endorsement by Nationwide Insurance. Each site is responsible for their own use of the concepts and examples presented.

First, a word from our announcer:

With a few exceptions, this is an overview! Where possible there are technical details you may be able to use. As you frequently hear when anyone asks for recommendations, “IT DEPENDS” is the answer and it applies here too. The information in this session is based on *our* experiences as long-time VM-ers building virtual Linux farms.

Interaction is good! Please ask questions whenever you want. We'll all get the most out of this session that way.

Agenda

- Why are we here
- A little background and foundational info
- Doc, it *hurts* when I do this! (Probable pain points)
- Potential ideas to reduce the pain
- Conclusions

Why are we here?

- I'm here to show what we've done with SAN on our zLinux servers
  - Why SAN, pain-points, and how we are overcoming them
  - A z/VM management perspective (not Linux nor Storage)
- I hope you are here because…
  - You are thinking about using SAN and want to know more
  - You already are using SAN and would like to know how to deal with pain-points
  - You had nothing better to do in this hour and love to hear me speak

Why are we here?

- I won't be giving away all my secrets
  - Meant as a discussion on our experiences, not technical details
  - You will have to put the pieces together
- BTW, I may make your head explode at some point
  - If not, you are napping

Why use SAN

- Main decision to use SAN at Nationwide on zLinux
  - Great for large filesystems
    - LVM'ing Mod 9s or 27s for 500GB+ filesystems was not optimal
      - *Takes at least 74 Mod 9s for a 500GB filesystem!*
  - Less space overhead (waste)
    - ECKD formatted DASD has a 4K block overhead
      - *Example: A raw Mod 9 has roughly 8119.63M (7.93G) of space, but formatted at 4K and it has 7043.20M (6.87G) usable space*
    - SAN devices do not have a formatting overhead
  - Other reasons… SAN used on other platforms, Linux SA group used to it, etc

Why not use SAN?

- DR considerations - complex
- No good way to manage luns
  - No built-in or vendor solutions
- Instrumentation lacking
- Unfamiliar to a mainframe person
- Because it isn't ECKD! (I'm kidding!)

In the beginning…

- Decision was to use non-NPIV for test/dev and NPIV for production
  - NPIV or Node Port ID Virtualization
    - FCP subchannel/devices are mapped only to specific target luns
    - Protects access from other servers/LPARs
  - Non-NPIV luns are masked to the FCP CHPID, so any device on that CHPID could access the luns
    - Be careful trying to share luns!
  - NPIV is a pain; Non-NPIV is easier but not as secure

Replication for DR

- Production (NPIV) luns replicated for DR
  - Targets would be on “fast spindles” and clones (think flashcopy) would be made to “slower spindles” for DR-testing
  - Introduced triple pain-points
    - NPIV specific information needed to be mapped for DR (targets) and DR-exercises (clones)
    - FCPs in DR environment for each server needs to be pre-mapped and allocated
      - *If they don’t match between prod and DR, more pain*
    - Process to map DR targets or DR-exercise clones in Linux
      - *Not a happy place; SAN is in fstab*

SAN Management Challenges

- Mapping all the wwpn information correctly (NPIV!)
- Keeping track of allocations
  - What's used, what's free, how many, how big, which FCPs?
- Reporting/querying who has what in use
- The big thing is the *amount* of data to maintain and the *relationships* for that data
  - wwpns are 16 characters each, so are the luns
  - NPIV: virtual FCP wwpns mapped to two target wwpns and two luns. Then multiply x 3 because of DR and DR-test!
- Small pain – translating "our terms" for "storage terms"
  - All depends on which side you are looking at (hba vs virtual wwpn)

A “simple” provision – Prod & DR/clone

Host	Frame	Tier	Conf/Repl	FA Port	FA WWPN	HBA WWPN	Device	LUN(h)	LUN(d)	Size(GB)
VN2H6D12	2006	1	RDF1+TDEV	06E1	50000972081F5915	c05076fc7d001bc8	1665	1	1	33.72
VN2H6D12	2006	1	RDF1+TDEV	11E1	50000972081f5929	c05076fc7d002d48	1665	1	1	
VN2H6D12	2006	1	RDF1+TDEV	06E1	50000972081F5915	c05076fc7d001bc8	17F9	2	2	33.72
VN2H6D12	2006	1	RDF1+TDEV	11E1	50000972081f5929	c05076fc7d002d48	17F9	2	2	
VN2H6D12	2006	1	RDF1+TDEV	06E1	50000972081F5915	c05076fc7d001bc8	19B8	0	0	33.72
VN2H6D12	2006	1	RDF1+TDEV	11E1	50000972081f5929	c05076fc7d002d48	19B8	0	0	

Host	Frame	Tier	Conf/Repl	FA Port	FA WWPN	HBA WWPN	Device	LUN(h)	LUN(d)	Size(GB)
VN4H6D12	2006	1	RDF1+TDEV	06E1	50000972081F5915	c05076fc7d001a48	1666	1	1	33.72
VN4H6D12	2006	1	RDF1+TDEV	11E1	50000972081f5929	c05076fc7d002bc8	1666	1	1	
VN4H6D12	2006	1	RDF1+TDEV	06E1	50000972081F5915	c05076fc7d001a48	17FA	2	2	33.72
VN4H6D12	2006	1	RDF1+TDEV	11E1	50000972081f5929	c05076fc7d002bc8	17FA	2	2	
VN4H6D12	2006	1	RDF1+TDEV	06E1	50000972081F5915	c05076fc7d001a48	19B9	0	0	33.72
VN4H6D12	2006	1	RDF1+TDEV	11E1	50000972081f5929	c05076fc7d002bc8	19B9	0	0	

Host	Frame	Tier	Conf/Repl	FA Port	FA WWPN	HBA WWPN	Device	LUN(h)	LUN(d)	Size(GB)
VN2H6D12-DR	2008	1	RDF2+TDEV	06E1	50000972081F6115	c05076fc7d804648	0B72	2	2	33.72
VN2H6D12-DR	2008	1	RDF2+TDEV	11E1	50000972081F6129	c05076fc7d801b48	0B72	2	2	
VN2H6D12-DR	2008	1	TDEV	06E1	50000972081F6115	c05076fc7d804648	1E56	4	4	33.72
VN2H6D12-DR	2008	1	TDEV	11E1	50000972081F6129	c05076fc7d801b48	1E56	4	4	
VN2H6D12-DR	2008	1	RDF2+TDEV	06E1	50000972081F6115	c05076fc7d804648	2721	0	0	33.72
VN2H6D12-DR	2008	1	RDF2+TDEV	11E1	50000972081F6129	c05076fc7d801b48	2721	0	0	
VN2H6D12-DR	2008	1	TDEV	06E1	50000972081F6115	c05076fc7d804648	2739	1	1	33.72
VN2H6D12-DR	2008	1	TDEV	11E1	50000972081F6129	c05076fc7d801b48	2739	1	1	
VN2H6D12-DR	2008	1	RDF2+TDEV	06E1	50000972081F6115	c05076fc7d804648	3C96	3	3	33.72
VN2H6D12-DR	2008	1	RDF2+TDEV	11E1	50000972081F6129	c05076fc7d801b48	3C96	3	3	
VN2H6D12-DR	2008	1	TDEV	06E1	50000972081F6115	c05076fc7d804648	3C98	5	5	33.72
VN2H6D12-DR	2008	1	TDEV	11E1	50000972081F6129	c05076fc7d801b48	3C98	5	5	

What does a Linux guest need for SAN?

- Actually, not that much!
  - Two FCPs attached to each guest
  - wwpn/lun information of the target storage

Device	Target wwpn	Target lun
0.0.0100	50000972081f6115	0000000000000001
0.0.0200	50000972081f6129	0000000000000001

  - Using zfcp_disk_configure (S390 tools package) to bring them online, and multipath tools to enable
  - Poof! (assuming they are masked right)

We use a PARMFILE to hold lun info

- We adopted the use of a CMS disk with “parmfiles” used to personalize the zLinux servers

```
HOST=nzvmds104
ADMIN=10.83.196.16
BCKUP=10.82.216.3
DRADMIN=10.121.196.6
DRBCKUP=10.120.216.138
VIP=
DRVIP=
ENV=PROD
BOOTDEV=01B0
LVM=/db2fs,10G,/db2logfs,10G
SAN_1=0100:50000972081f5929:0000000000000000,0200:50000972081f5915:0000000000000000
RDF_2=0100:50000972081f6115:0000000000000000,0200:50000972081f6129:0000000000000000
CLO_3=0100:50000972081f6115:0001000000000000,0200:50000972081f6129:0001000000000000
DR_EXERCISE=no
```

Easy, right?

- For smaller shops, maybe it is not that bad
  - A handful of servers using SAN is manageable
- Medium shops will begin to feel the pain points
  - Couple dozen or more servers will begin to be interesting, but still reasonably managed with the right tools/processes
- Larger shops are in for a fun time
  - Hundreds of zLinux servers with Prod/Test/DR/DR-test environments
  - Keeping it all straight and working cleanly is complex at best (Read “job security” ?)

Attempt to manage

- Tried a spread-sheet first
  - Worked out okay for the first dozen or so servers
  - Prod server info (NPIV, DR, DR-test) started getting a bit hairy
  - Too much cut/paste to get info to z/VM and/or Linux
- Moved it to a z/VM text file with a little REXX
  - Worked for a short time longer (no cut/paste)
- What about a CMS NAMES file?
  - That's the ticket! It should help with keeping things straight

Attempt to manage – CMS NAMES file

CMS NAMES file simply a tagged data file

:vmid.SZVJT014
:assigned.2011-02-08
:hostname.szvmjt014
:type.inuse
:side.1
:node.VST
:chpid.50
:rdev.0105
:virtwwpn.
:sanframe.2008
:sandev.22ed
:targwwpn.50000972081f6115
:lun.025900000000000
:size.8.43
:uuid.
:sanalias.zlinux-ps1
:tier.1
:srdf.tdev
:fasp.06e1

Attempt to manage – CMS NAMES file

- Great idea **J**
- Tools available (NAMEFIND) to look up data in all kinds of different ways
- Tools NOT available to update the data easily
  - Wrote CMS Pipelines, Xedit macros, etc to manage

Attempt to manage – CMS NAMES file

- Allocate new luns: SANALLOC SZVJT999@VST 8 2

RDEVPAIR: rdev 0106 node VST type CHAN
RDEVPAIR: CHPID 50 side 1 CEC p5
SANALLOC: 2 channels or HBAs or FCP adapters
GETSANPAIR: looking for size 8.43 on node VST
GETWWPNS: looking on node VST
using rdev 0106 pair 0206 for SZVJT999 at VST side 1
using rdev 0206 (pair) for SZVJT999 at VST side 2
GETSANPAIR: using frame 2008 device 2319
SANALLOC: 2 paths through the fabric to each LUN
GETSANPAIR: looking for size 8.43 on node VST
GETWWPNS: looking on node VST
GETSANPAIR: using frame 2008 device 231a
SANALLOC: 2 paths through the fabric to each LUN

Attempt to manage – CMS NAMES file

- Querying: QSAN USER SZVJT999

FCP 0106 dedicated to SZVJT999 on VST
    WWPN 50000972081f9115 LUN 0285000000000000 size 8.43 type pri
    WWPN 50000972081f9115 LUN 0286000000000000 size 8.43 type pri
FCP 0206 dedicated to SZVJT999 on VST
    WWPN 50000972081f9129 LUN 0285000000000000 size 8.43 type pri
    WWPN 50000972081f9129 LUN 0286000000000000 size 8.43 type pri

Attempt to manage – CMS NAMES file

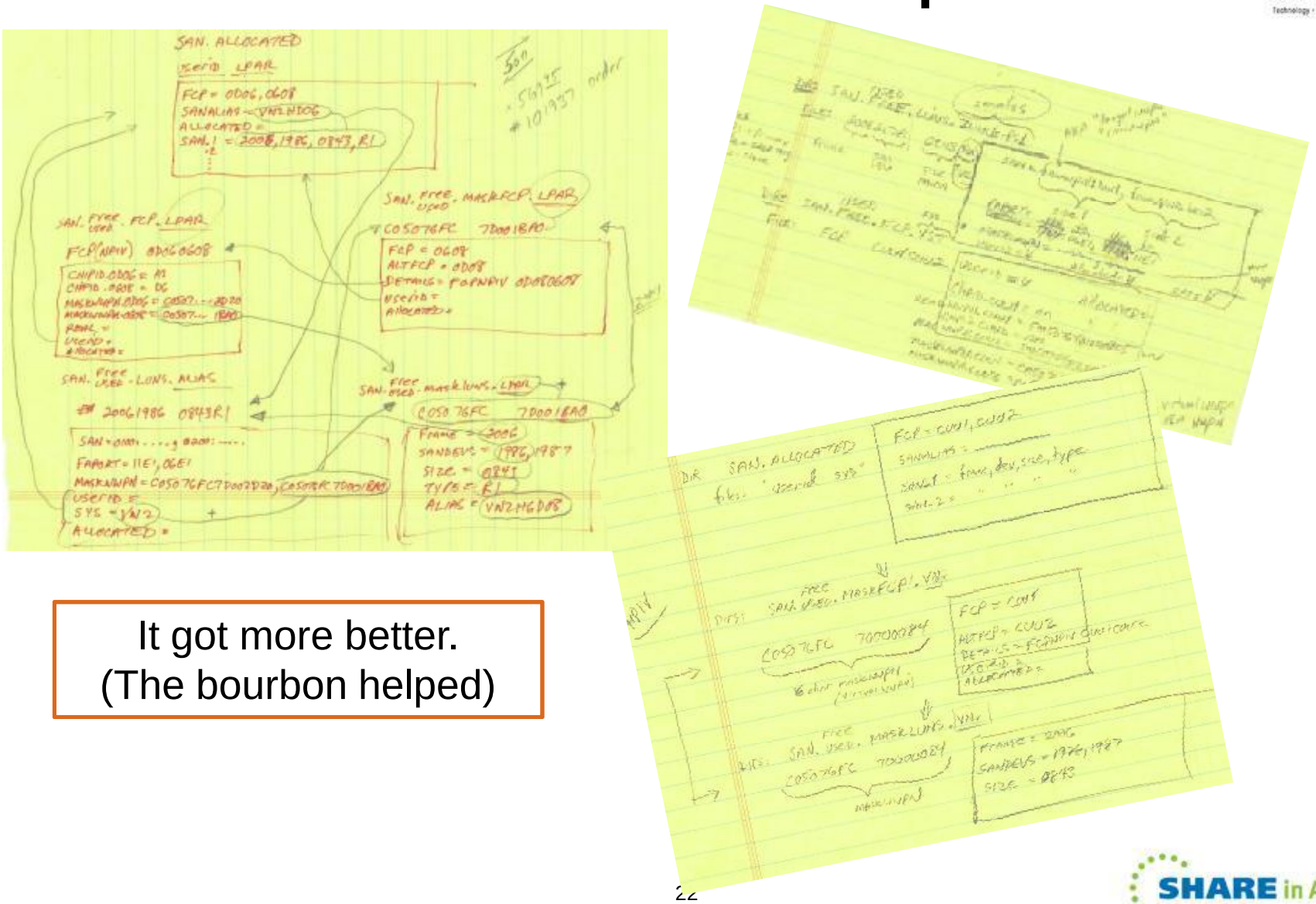
- Querying/reporting: QSAN WATCH

```
   1 LUNs of   33.72 in South
shortage: 1 LUNS at 33.72 on SOUTH
  70 LUNs of    8.43 in South
Avail FCP pairs: ( VSU:60 VS1:32 VS2:252 VS3:30 VS4:236 VS6:306 VS8:310 )
   8 LUNs of   33.72 on VN2
  24 LUNs of    8.43 on VN2
   8 LUNs of   33.72 on VN4
  24 LUNs of    8.43 on VN4
   8 LUNs of   33.72 on VN6
  26 LUNs of    8.43 on VN6
   8 LUNs of   33.72 on VN8
  20 LUNs of    8.43 on VN8
Avail FCP pairs: ( VNU:48 VN1:48 VN2:204 VN3:48 VN4:206 VN6:205 VN8:194 )
```

Attempt to manage – CMS NAMES file

- Things worked okay, except
- Not relational data
  - Keeping all the data “chained” turned into a nightmare
  - Things “break” a tad too easy if something hiccups
- Adding new FCPs, new luns to the NAMES file is difficult
  - Again, querying is pretty easy – updating is not
- Turned into “Not so great an idea” **L**

I had an idea that started on a napkin…



It got more better.
(The bourbon helped)

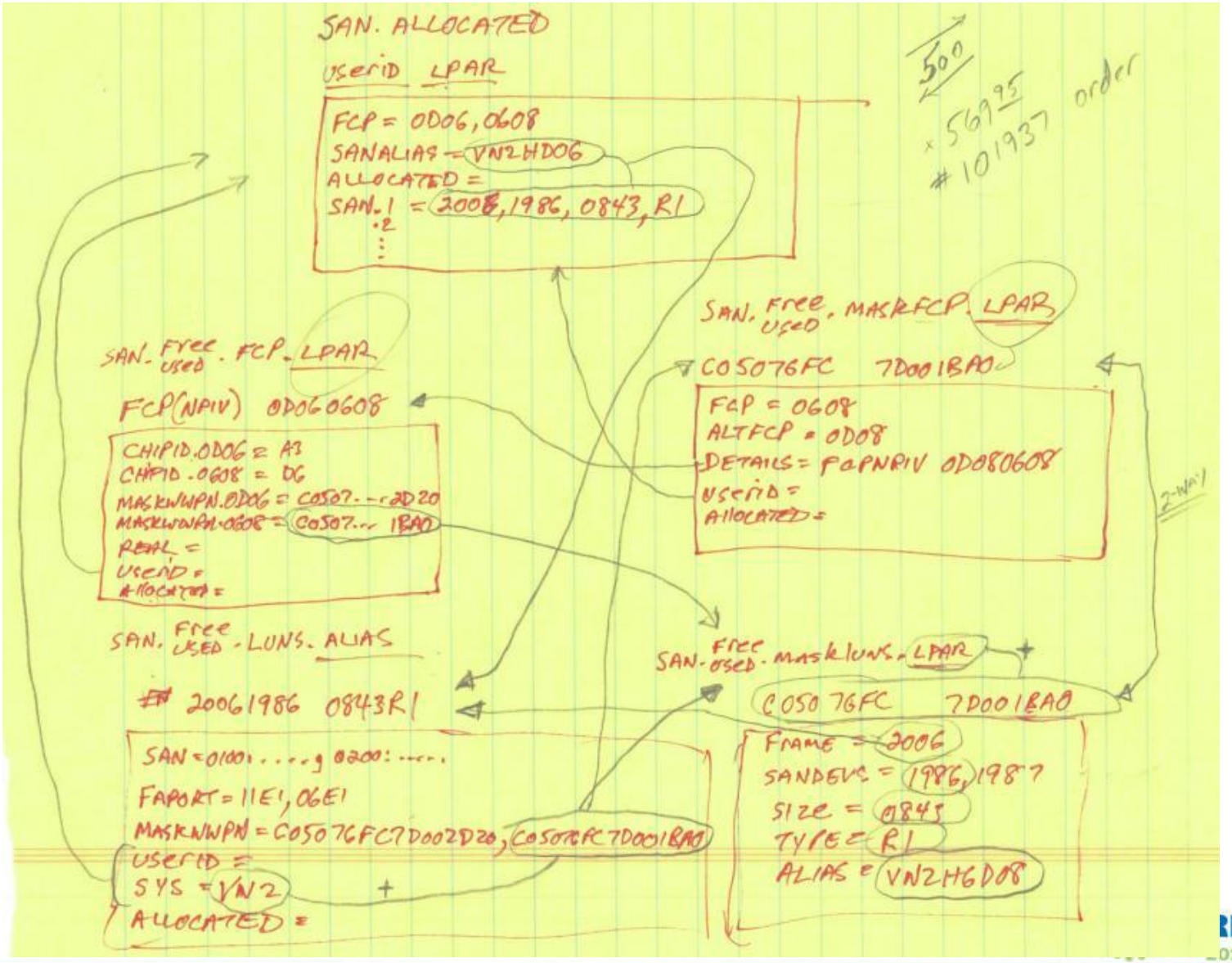
Relational data structures!

- The key was to build relational data structures
- I chose to do it in VM with native tools
  - Text files with tagged data in SFS directories
  - SFS (Shared File System) allows access across LPARs
  - Text files keep it simple to write tools to manipulate
  - Tags keep it simple to build relationships & find info
- You could do it off-platform
  - A MySQL solution on a Linux guest would be fine
  - Would need to build a way to get the data easily from VM

A relationship example

- Storage folks assign a “host” to the luns, but we have pools to provision luns as-needed, so the host is an “alias”
- An alias would map to SFS directories of:
  SAN.FREE.LUNS.alias (luns available to the host)
  SAN.USED.LUNS.alias (luns allocated to the host)
- In those directories *may* be files (fn ft) like
  20061986 0843R1 (frame, device + size, config)
- The content of that file would be
  SAN=0100:wwpn:lun,0200:wwpn,lun
  MASKWWPN=wwpn,wwpn…
  USERID=NZVJT999
  SYS=VN2
  ALLOCATED=yyyymmdd

Remember the picture?



Easier than it looks, really

- Once the structures are built, it should make more sense
- The structures, directories and files make it easier to manage
  - REXX & CMS Pipelines easily find and manipulate the data
  - Adding/removing relationships (provision / decomm) is almost as easy as moving a file from one directory to another and updating a couple of tagged items
- Tools make it easy to report on usage, what's available
- Simple and fast queries, esp. good for crit-sits
  - Lost fabric path, what servers affected?
  - Device errors, what server(s) have those luns?

Yeah, but what about DR?!

- Processes in place to handle DR setup
  - Take the production CP directory
  - Query the server's FCPs for the remote LPAR
  - Update directory info and transmit to other LPAR
  - On target DR LPAR, load into CP
- Remember the PARMFILE?
  - That stays the same, just maintained and built easier
  - Linux runs a boot script to know that it is in DR mode (LPAR) or DR-EXERCISE mode (flag) and picks RDF or CLO entries
    - Reloads the luns into multipath
    - As long as replication/cloning worked, that's it!

What about EDEVs?

- EDEV: emulated device that represents a real SCSI device
- EDEV could help with a lot of the pain(s) since they are treated like minidisks by VM
  - Can carve into smaller minidisks
  - Can use for paging/spooling too
- Cons of EDEV:
  - Slower Throughput (protocol translation)
  - Increased Overhead (hypervisor CPU)
  - Because IBM recommends direct FCP – at least they used to
  - A little too buggy for use when we used it
    - Worth looking at again though!

Conclusions

- SAN is certainly a good thing in zLinux
- You need to manage the required data carefully, especially with NPIV and DR requirements
  - You may find yourself having to build something
  - Be patient; it will be daunting at first
- Make friends with a storage support person at your location – it will pay off!
  - Learning their lingo and they learning yours will help a lot
- EDEVs are something to look into
  - Test them well; get IO stats between FCP attached SAN, EDEV, and ECKD
  - TEST failures and make sure the system and zLinux recover as you would expect them to

Conclusions

- Heard about the next z/VM release yet?
  - What will SSI clusters do for your zLinux guests w/SAN?
    - Live Guest Migration – same thing
  - NPIV and dedicated FCPs will be things to consider

Contact info

Light travels faster than sound, that's why people seem bright until you hear them...

James Vincent
Sr. z/VM Systems Programmer

Phone: (614) 249-5547
Internet: James.Vincent@nationwide.com