Linux on VM
From Woe to Whoa!
Part I - How did we get here, Toto?

Session 9212

Jim Vincent

James.Vincent@nationwide.com

March 7, 2006

SHARE 106

Overview and Disclaimer

Disclaimer:

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.

Overview:

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"!

The information in this session is based on our experiences as long-time VM-ers adding virtual Linux.

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

Topics

- Our Linux Decision History
- Our Environment
- What do we expect Linux to do for us
- Direction – What drives our project
- A learning process
- Conclusions

Our Linux Experience

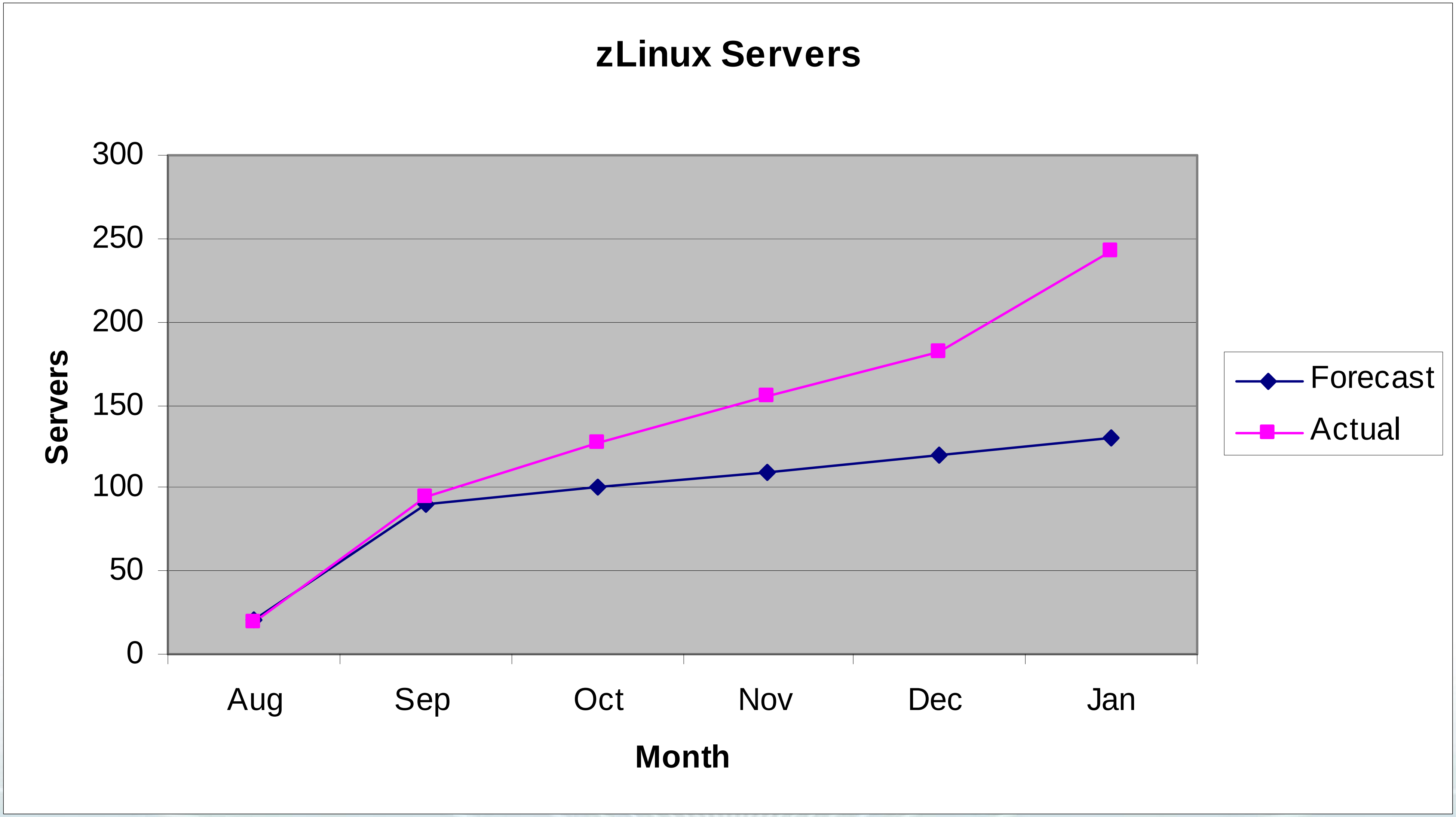
Our Linux Decision History
The story of Woe

- 2000 – Marist Distribution (based on Red Hat)
  - First offering of install lab at SHARE
  - Built one in-house to play with
  - Wrote up recommendation to management; Little interest or direction
- 2002 – SUSE 7
  - Basic demo of Apache and Samba
  - Wrote up recommendation to management; Little interest or direction
- 2004 – Red Hat
  - Intel, pSeries and zSeries pilots planned and started
  - zSeries waned quickly and work ceased

Our Linux Decision History
The story of Whoa!!

- 2005 – The fun begins!
  - New Emphasis on virtualization
    - Fasten your seat-belts!
- Proof of concept system originally had three small business applications
  - Then we had at least seven with more wanting on ASAP
- *Our* initial thought was to tackle File/Print sharing
  - Naaaw! That’s too easy – we started with J2EE servers!
  - WAS, WAS Portal, IHS, DB/2, etc.
    - The Hoover’s of the zLinux workload
- Anticipated having about 120 total servers by year-end 2005
  - It is growing faster than anyone thought it would

Be careful what you ask for! J



Our Environment

Environment

- Before we got serious about Linux
  - 3 z900 processors; mostly z/OS; models 104, 107, 1C8
    The 104 had 24GB of storage and ran:
    - § 4 z/OS LPARs
    - § 1 Coupling Facility LPAR (ICF)
    - § 2 relatively small z/VM LPARs which use about 200+ MIPs
      - LPAR 1: 3 shared CP; 3 GB Central storage; 1 GB Expanded storage
      - LPAR 2: 2 shared CP; 768 MB Central storage; 256 MB Expanded storage
      - Mostly web services
      - Some application development, support and cooperation with z/OS
      - Business Recovery support

Environment

- For Linux **pilot**
  There was capacity to create another small z/VM LPAR on the 104.
  - 3 dedicated IFL engines
  - 8 GB memory
    - 6 GB Central; 2 GB Expanded

- **Today** – 2 new z990s dedicated to Linux
  - 5 IFL engines for test/dev and **4** IFL engines for production
  - 64GB and 56GB memory respective
  - 4 z/VM 5.2 LPARs on each
    - 1 additional test LPAR on development box for sandbox
    - 9 total LPARs
  - **Growing FAST!**

Our slogan…

Nationwide

Nationwide®

On Your Side℠

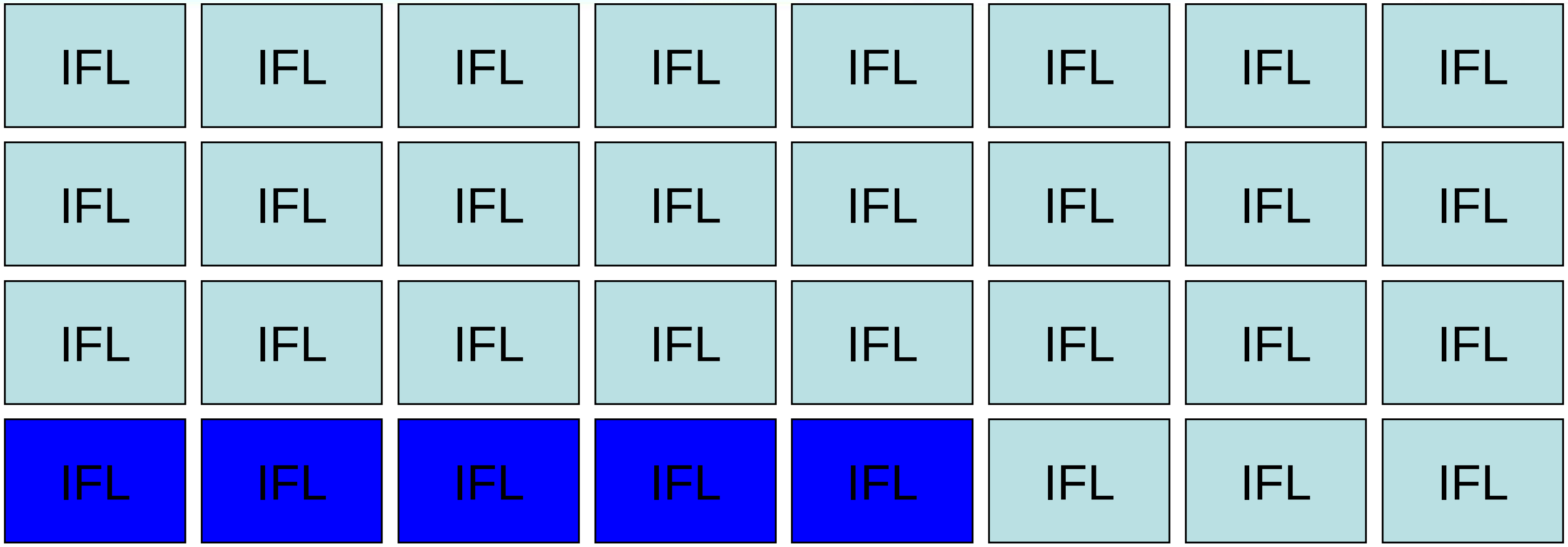
LIFE COMES AT YOU FAST®

“Our” slogan…

TUX COMES AT YOU FAST

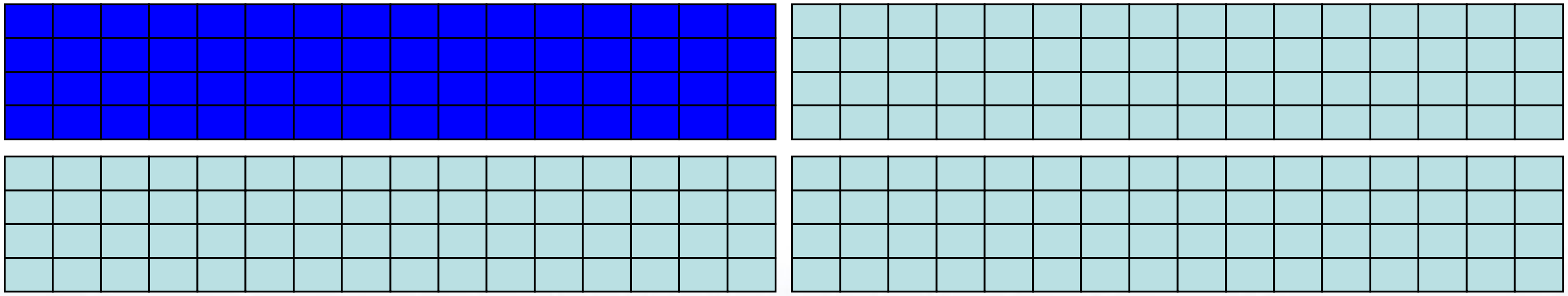
IBM z990 Platform (test/dev)

Processors
3 IFLs
Max 32
(16%)



Max MIPS on the z990 is >10,000. zSeries Linux has a potential to out-MIP the z/OS & z/VM traditional z environments very soon!

Memory
64GB
Max 256GB
(25%)



Network
5 OSA Cards
Max 48
(10.4%)

What Do We Expect Linux to Do For Us?

Problems to solve

- Server Proliferation
  - Space that previously was required to house a few mainframes is now mostly consumed by multitudes of all type of servers, network hardware, other support hardware
    - Sun, HP, multiple brands of Intel
    - Routers and switches
    - SAN, NAS, data warehouse, etc

Problems to solve

- Provisioning
  - Many requirements for stand-alone server
    - Order and obtain hardware – several weeks
    - Physical install
    - Optional external disk subsystem configuration and connection
    - Network configuration and connection
    - OS load
    - Middle-ware load
    - Application load
  - Many hands and significant time
    - Usually would take several weeks (6-8 at least) or more before the customer would get the box

Vision and Expectations

- Physical space and environmental reduction
  - One z990 IFL engine can support 10-30 (or more) virtual servers
  - A z990 can have up to 32 IFL engines so it *could* replace 300+ servers
  - Significant savings in physical space, power, cooling
- Reduce network complexity
  - A small number of physical network connections (OSAs with VSWITCH) can support all of the virtual servers in contrast to every stand-alone server having 2 or more interfaces it must manage
- Quicker provisioning
  - Setting up new server can be as fast as your disk copy tool
    - § Depends on software needed on server and amount of manual effort

Direction – What Drives Linux and Virtualization on zSeries

Direction – getting Linux on z rolling

- Start with technicians then try to influence organization
  - Common with existing zSeries shops; especially those with z/VM already in their shop ("skunk works")
    - Build something and demonstrate function and don't bother to tell anyone what it is and where it is running
    - “We will build it and they will come”
    - Be prepared to have the idea crash and burn when presented to management
  - Challenges
    - Organization barriers – turf wars
    - ‘Opinions’ used instead of good technical evaluations and decisions
    - Workload – real work vs. “fun” stuff like Linux
- Start with CIO (upper management) and direct organization
  - More common as industry accepts zSeries virtualization solution
  - Driven by business need (e.g. space restraints, rapid growth, etc)
  - Typically causes more structured implementation and wider acceptance
    - Some will still kick and scream, though not real loud

A Learning Process

Learning – everyone has to!

- Mainframe methodology differs from non-mainframe methodology
  - Repeatable automated processes versus hands-on "hacker" install
    - Typical mainframe person accustomed to well-documented, repeatable processes that permit automation for multiple installs
    - Preference for install once and copy rather than repeated installation
    - Expect software to be installed in one location and configuration in a common location
  - Different philosophy for management:
    - Privilege levels differ for installing OS, installing middle-ware, configuring
  - Different maintenance philosophy
    - Expect to regularly upgrade software on all servers

Distributions

- Documentation differs greatly
  - Red Hat
    - Installation instructions begin at loading the RAM disk into memory
    - It appears to have been an afterthought
  - SUSE
    - Shows how to build the virtual server directory and copy the RAM disks to VM
    - It seems to understand the zSeries and z/VM environment
  - Both
    - Incomplete (inadequate) documentation of install parm information for all environments
      - Some not documented
      - Little more than syntax
      - Incomplete or no examples

Distributions

- Media
  - CDROM
    - Can't load directly on z/VM
    - Make distribution media available on existing Linux (or Unix) workstation
    - Make RAM disk images available to FTP to VM for install
      - Directly from CDROM
      - Extract from mounted ISO file

Distributions

- Default package list
  - Red Hat
    - § Large list of packages in minimum load
    - § Security template required omitting load, turning off or disabling many packages
    - § Runtime compatibility for 31-bit not included in default 64-bit load
  - SUSE
    - § Smaller list of packages in minimum load – basic runable Linux
    - § Security template required only a handful of changes

Distributions

- Red Hat Enterprise Linux AS
  - When installing Red Hat Enterprise Linux AS 3 64-bit, default RAM disk size was too small to build a complex DASD / LVM configuration.
    - Resolving this took several days and knowledgeable Linux ‘experts’ to identify. (Red Hat says this is fixed in AS 4)
  - Working bugs out of kickstart was a time-consuming repetitive process.
    - We started with a working kickstart script from Intel.
    - It was difficult to identify packages that are not on the s390 and s390x CDROMs.
  - Install (using kickstart) formatted DASD one-at-a-time (serially) (This may be a restriction of the kickstart process.)

Distributions

- SUSE Linux Enterprise Server
  - Install processes formatted multiple DASD in parallel
  - Never completely got AutoYaST to work
    - Realized that cloning / copying servers makes this less important

Linux Basics for z/VMers

- Even z/VM Sysprogs need to understand what Linux is up to.
  - What we know about running VM applications has an impact on how Linux is built on VM and knowing Linux to some degree helps get the points across to the Linux admins
  - Apply some mainframe disciplines / history / concepts to virtual Linux
- Learn how new devices are added, defined and identified in Linux for the distribution you are using.
  - Learn LVM too
    - § Differs depending on distribution (kernel level and/or LVM level)
    - § Using CKD DASD for virtual servers is likely to cause you to use it.

Linux Basics for z/VMers

- Take careful notes about what you learn so you can use them later
- Pick up on Linux tips/tricks to make your life easier (refer to note 1).
  - Something as simple as adding a "&" to the end of a dasdfmt command lets you run things in the background and not have to wait for them to complete to do something else.
  - RPM - learn how to search for packages, display information on them and how to install/uninstall them
  - Learn the file attributes and what they mean, along with the decimal representations of them.
    - For chmod commands, you need to understand them
  - know how to use tar and gzip
  - Keep cheat sheets on VI (VIM), ED, and one on Linux System Admin (http://www.cactus.org/~dak/sysadmin.html for example)
  - Symbolic Links are useful to know and use
  - killproc is your friend.   Know how and when to use it
  - You can create environment variables for common locations you "cd" to often. For instance, if you always cd to "/opt/var/html/web/tsm/www/logs/public" you can create an environment variable called $TLOG and then cd $TLOG to get there. Using the "tab" trick is okay, but it still takes longer than typing "cd $xxx..."

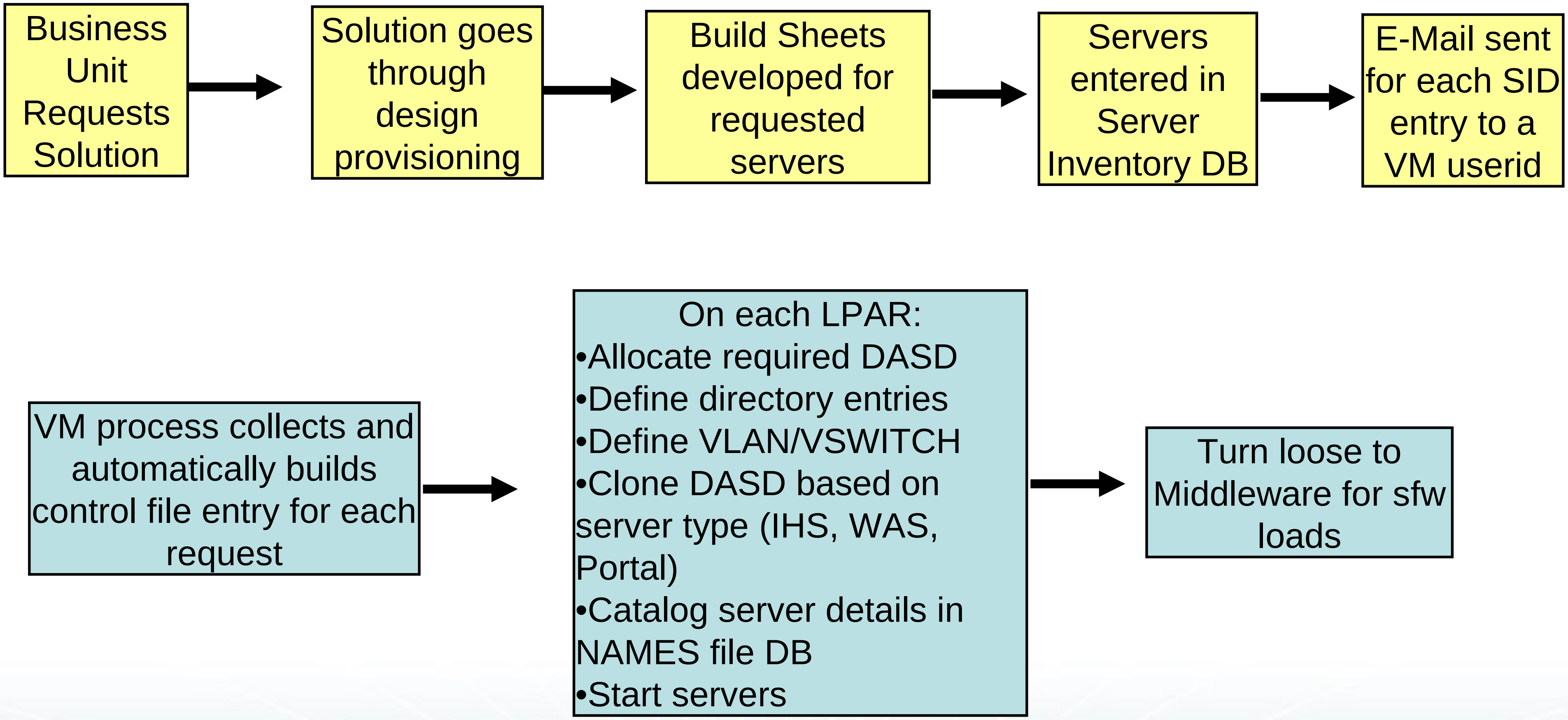
Cloning servers

- Cloning:
  - There are numerous ways to clone Linux images
    - PICK ONE and stick to it
    - Once you start using it, switching to a different way will be time consuming
  - If you have hardware disk duplication available (IBM Flashcopy, STK Snapshot), it can be a huge benefit to cloning
    - If not, DDR has to be used (slow)
  - You may want to create "standby" Linux images for quick deployment
    - If you have standard templates for the Linux servers, build a few extras and deploy them as requested
    - Makes you look like a wizard when someone asks for a server and 2 minutes later they can log on to it!

Cloning servers – my take

- Until a vendor solution is obtained, I rolled my own
  - Dirmaint used for directory maintenance… *sigh*
  - DDR for large-volume copy due to storage vendor choice and no tool support for z/VM
  - Multi-stage process so that things can be fixed if they have a glitch
    - *Otherwise known as a “finger check” in the control data*
- A server can be built in < 30 minutes
  - My personal best is 28 WAS/Portal servers in 1 hr. 15 mins.
- A picture may tell it best…

Cloning / server builds



Business Unit Requests Solution

Solution goes through design provisioning

Build Sheets developed for requested servers

Servers entered in Server Inventory DB

E-Mail sent for each SID entry to a VM userid

VM process collects and automatically builds control file entry for each request

On each LPAR:

- Allocate required DASD
- Define directory entries
- Define VLAN/VSWITCH
- Clone DASD based on server type (IHS, WAS, Portal)
- Catalog server details in NAMES file DB
- Start servers

Turn loose to Middleware for sfw loads

Our server build SLA is 10 days; goal is 72 hours from initial request to fully loaded

Linux workloads

- Linux on zSeries virtual servers *may* be able to run with small(er) memory (storage) sizes
  - "It depends" on what will run there
  - A basic Linux virtual server can easily run on 64MB of memory
  - An IBM HTTP Server can probably run in 128-256MB depending on the number of static pages and CGIs, etc
  - An IBM WAS Server probably needs 512MB-1GB
  - An IBM WAS Portal or DB server probably needs 2G or more
  - There are a lot of simpler application options!
    - Firewall
    - DNS
    - Web server
    - File and Print serving

Linux workloads / applications

- *Any* virtualization brings out the best and worst of applications
  - Bad things shine like the sun when they are virtualized
    - § Memory leaks
    - § Spin loops
    - § Poor design / configurations
    - § Logging and debugging options
    - § Intense computations
- Fixing any issues results in a much tighter, better performing application
  - And you can put more than one of them on a single virtual Linux server too

zSeries – is it memory or storage??

- zSeries Hardware
  - CPU
    - Effective speed much higher than raw speed
  - Storage (aka memory)
    - Maximum on a 32-engine z990 is 256 Gigabytes
      - A z9 goes up to 512G – but is that still enough?
    - Fixed allocation to each z/VM LPAR
    - z/VM dynamically allocates to virtual servers
  - DASD (aka disk or storage)
    - Count Key Data (CKD) – traditional zSeries
    - SAN – used on many stand-alone servers

zSeries networks

- zSeries Hardware
  - Open System Adapter (OSA) Express 2 Gigabit Ethernet
    - Gigabit adapter with a smart network controller
    - zSeries LPAR microcode allows:
      - Sharing of the same OSA across LPARs
      - Multiple Read/Write/Data groups to be attached to virtual server or defined as a VSWITCH
  - **Tune in to Part II for *far* more details…**

"Experts"? I’ll take two…

"My definition of an expert in any field is a person who knows enough about what's really going on to be scared.“ - PJ Plauger

- "Experts": Do they really exist?
  - There are many people with varying levels of experience in specific areas
  - There are few (if any) who know enough about everything
  - Make friends with people who have knowledge in:
    - Mainframe disciplines
    - Linux
    - Network
  - Learn as much as possible about all of these
    - Or at least learn how to contact them when you need to!

Conclusions

Conclusions

- Linux definitely is Linux – the same on all platforms
  - Code written for Linux on any platform can usually be used on any other platform that supports Linux with only a recompile (usually)
- Linux virtualization on zSeries can and does:
  - Reduce cost
    - Just software costs per engine can save you BIG $$
  - Reduce complexity
    - Sharing R/O DASD, less complex network (wires), etc
  - Accelerate provisioning
    - Feasible to provision servers in minutes
  - Reduce human error of manual installation, configuration and even patching

Conclusions

- Not every workload is suited to Linux on zSeries
- Not all software is ready for Linux on zSeries
- Things are changing rapidly

- Be careful what you ask for because you may get it!

- …story continued in the next session!

Contact Info

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

Jim Vincent

Systems Engineering Consultant (whatever…)

Phone: (614) 249-5547

Internet: James.Vincent@nationwide.com