zPDT and RD&T User Experiences: Running z/OS

Stephen Norris
CA Technologies

Monday, March 2, 2015: 1:45 PM-2:45 PM
Session Number 16642
About the Author

• Senior Director, Engineering at CA Technologies
  • Systems and Performance Management
• 23 Years in Computer Software Development
• Lead architect for zPDT environment at CA Technologies
• Co-authored IBM virtualization book with Ken Barrett
  • Running Mainframe z on Distributed Platforms: How to Create Robust Cost-Efficient Multiplatform z Environments
Initial Configuration/Usage

• Single Developer
  • One user
  • Local login
  • Single, native z/OS system
• Linux Configuration
  • Laptop
  • Single hard drive
Problems with Initial Configuration

- No remote access
  - Possible to use VNC/Remote Desktop
    - Response time was poor
    - Would require additional effort to serialize access
- Little configuration flexibility
  - Many users required access to multiple systems
  - Access to the mainframe data was often required
- Hard to collaborate with single localized login
  - Many teams have users across the globe
- Requires all user to have advanced Linux and System z knowledge
The Next Step

• To overcome several of the initial obstacles
  • The Linux host was moved from a laptop to a server
    • Faster/more processors
    • More memory
    • Increase storage capacity
  • Multiple environments could be located in a single site
    • Allowed localized expertise to be used
    • Simplified system maintenance
  • z/VM was installed as a host for three z/OS guests
    • Allowed for greater flexibility in the environment
      • Multiple concurrent users
      • Ability for testing across multiple systems
The Next Step (cont.)

- Networking was installed on z/VM and z/OS systems
  - Created the ability for simultaneous remote access
  - Allowed communication with other network resources
  - Enabled user to move data between the zPDT and mainframe
- Virtual networking switch was created on z/VM
  - Simplified networking environment to support multiple guests
  - Minimized the hardware requirements for the server
The Next Step - Configuration

Physical Server

VM01

VM11  VM12  VM13

VMSWITCH  Address 0404

RAC

Path F0 NIC 1

Linux Host

Path F1 NIC 2

LAN

Complete your session evaluations online at www.SHARE.org/Seattle-Eval

Copyright Stephen Norris © 2014
Limitations of “The Next Step”

- Still using ADCD
  - No simple method to update the System z software
  - Version/Level support was limited
- Still no data sharing between guest systems
- Environments were slowing down as work was increased
  - More work was creating system strain
  - Users were demanding more responsive systems
- Users required 24/7 access
  - Needed to ensure system reliability
  - A dependable backup solution was required
  - Hardware tolerance needed to be considered
Evolution of a New Environment

- Hardware requirements revisited
- Using mainframe environment as a template
- Resource sharing investigated
- Birth of the DASD Repository
- Deployment strategy created
- Backup/Restore reimagined
- Emergency Recovery System
Hardware Requirements Revisited

- More and Faster Processors
  - While z Environment is limited by the licensed dongle, extra processors can be used by the Linux host
  - Faster processors provide more cycles for emulated systems
- Upgrades to Linux server hard drives
  - Upgraded to 15K RPM from 7.2K RPM
  - Configured in RAID 5 for fault tolerance
  - Additional storage added for future expansion
Hardware Requirements Revisited (cont.)

- More memory
  - Paging in the emulated System z environment dramatically reduces performance
  - New servers were ordered with 64GB or more of RAM
  - Allocations to each z/OS guest designed to minimize paging

- Hard drive configuration
  - 15,000 RPM drives
  - All but one configured in RAID 5
  - Single drive configured as RAID 0
Using Mainframe Environment as a Template

- CA Technologies has several virtualized systems
  - Each is based on a template
  - Improves maintenance cycles
  - Reduces administrative overhead
- New zPDT z/OS environments are based on this template
  - Provides familiar look and feel to existing users
  - Reduces overhead of maintaining new zPDT environments
    - Copy system packs from mainframe to use on zPDTs
    - System changes enacted on mainframe can be easily ported
    - Provided a mechanism to apply IBM APARs to zPDTs
Resource Sharing Investigated

- Initial configurations had three independent z/OS guests
  - Provided redundancy
  - Not a typical IBM environment
  - Created a lot of maintenance overhead
- Best environment would allow the sharing of resources
  - Datasets, catalogs, etc
  - At this time, IBM released support for SYSPLEX on zPDT
- The base environment was converted to use SYSPLEX
  - The new environment was quickly rolled out
  - Consisted of:
    - 2 virtual Coupling Facility machines for redundancy
    - 3 z/OS guest systems

Complete your session evaluations online at www.SHARE.org/Seattle-Eval
Birth of the DASD Repository

- Needed somewhere to store stable base configurations
- Also assembling a core group of reusable DASD
  - Initialized empty volumes
    - Eliminated repetitively creating and initializing new DASD
    - Base library of various models of DASD
  - DASD copied from the mainframe
    - Now stored in a central location
    - Downloaded once, available to any server
- DASD with non-volatile data
  - Stored on one server
  - Can be mounted on a read-only NFS share
  - Connected to many zPDT environments
Birth of the DASD Repository (cont.)

Mainframe zEnterprise
- Regularly Updated Software
- Accredited Service Levels

DASD Repository

Server
Server
----
Server
Server
Deployment Strategy Created

- New environments were being requested
  - Required the creation of new configurations
  - Required retooling of old environments
- Base image was created
  - All DASD for z/VM and the z/OS guests
  - The device map file
  - Backup scripts
- New base image could be rolled out via
  - Flash drive
  - FTP server
  - File share
Deployment Strategy Created (cont.)

- Deploying a new server was reduced to:
  - Creating a new Linux host
  - Deployment of base image
  - Customization of base image
    - New System z host names
      - Simplified by the use of symbolics and symbols
      - Most of the changes were on z/VM
    - Network configuration changes
    - Customization of backup scripts (if needed)
Backup/Restore Reimagined

- To provide data integrity, the systems and emulator are shutdown during the backup process
- Initially ran backups of the full environment
  - Used gzip to compress individual DASD files to USB drive
    - Could not find utility that was faster
    - gzip commands were scripted
    - Took over a day
- Reevaluated backup strategy
  - Backup only DASD files with volatile data
  - Initial backup is to local hard drive to improve performance
    - Provided a 75% reduction in backup times
  - Local hard drive backup files are copied to NAS devices
Emergency Recovery (ER) System

- Sometimes modification of IPL parameters disables the IPL of the system
- IBM provides a utility to edit datasets from the Linux host
  - This step is complicated
  - Requires knowledge of exactly what the parameter needs changed
- A better option is to have an ER system
  - Runs on its own system packs and IPL libraries
  - IPL is independent of the development system packs
  - Has access to configuration packs of the development systems
Emergency Recovery System
Questions?

Complete your session evaluations online at www.SHARE.org/Seattle-Eval