Linux Servers on System z: Benefits and Features of Virtualization in the Enterprise Data Center

Rick Barlow
Nationwide Insurance

August 3, 2010

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 my experiences as a long-time VM-er adding virtual Linux.
Interactive is good! Please ask questions. We'll all get the most out of this session that way.

Topics

- Our Environment
- Simple “Logical” TCO - Why Virtualize Hardware?
- Virtual Networking
- High Availability
- Disaster Recovery Enablement
- Performance
- Conclusions

Our Environment

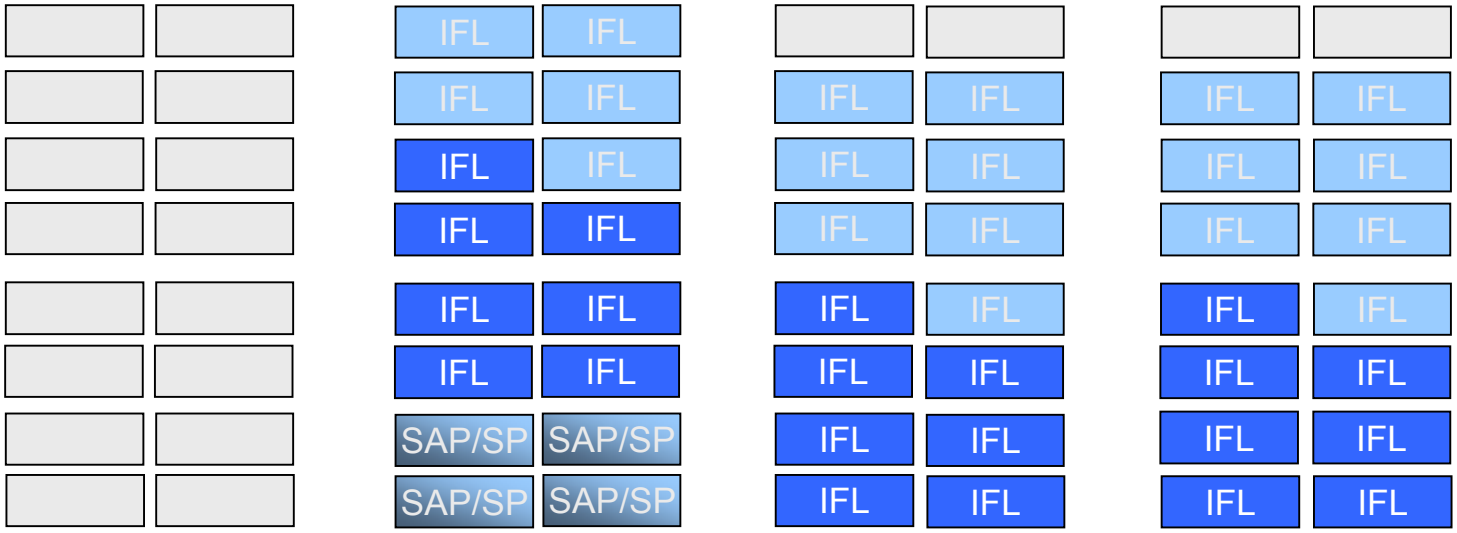
- Dedicated machines running z/VM and Linux only
- **T**wo z990 installed in 2005
  - Development box
    - 5 IFLs - grew to 8 and then to 16
    - 64GB memory - grew to 120GB
    - 5 z/VM LPARs (sandbox LPAR for system programmer test)
  - Production box
    - 3 IFLs - grew to 7 and then to 15
    - 56GB memory - grew to 112GB
    - 4 z/VM LPARs
- Upgrade to two z9 in November 2006
  - Development box
    - 8 IFLs - grew through several upgrades to 20
    - 128GB memory - grew through several upgrades to 352GB
  - Production box
    - 7 IFLs - grew through several upgrades to 38
    - 128GB memory - grew through several upgrades to 240GB

Environment

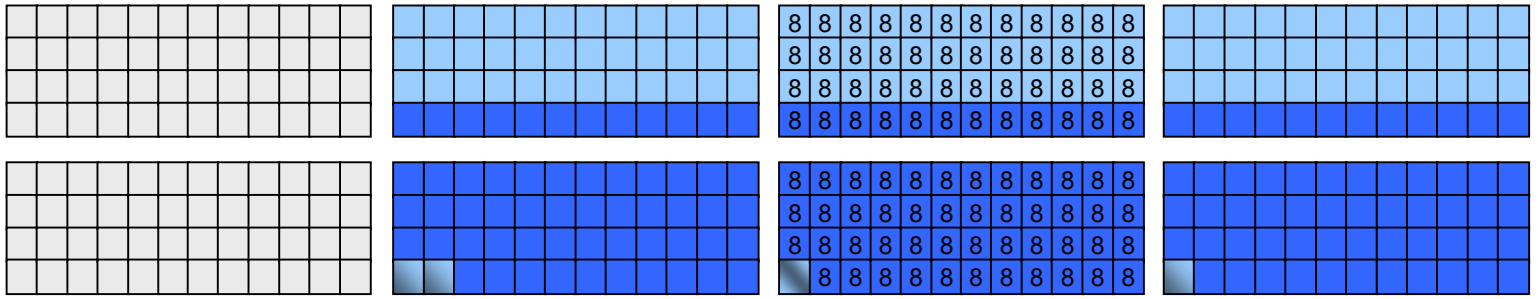
- Upgrade to two z10 in January 2009
- Today
  - Development box
    - 21 IFLs
    - 480GB
  - Production box
    - 33 IFLs
    - 352GB
- **Growing FAST!**

IBM z10 Platform (development/test)

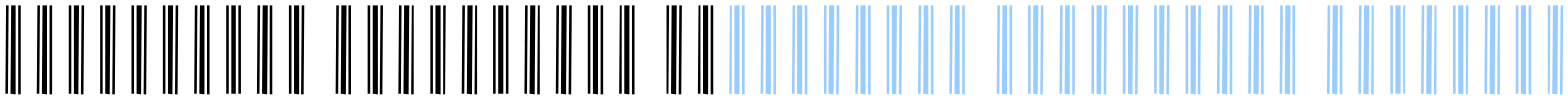
Processors
21 IFLs
Max 40 (3 books)
(53%)
Max 64 (4 books)
(33%)



Memory
480GB
Max 1136GB
(42%)
Max 1520GB
(28%)



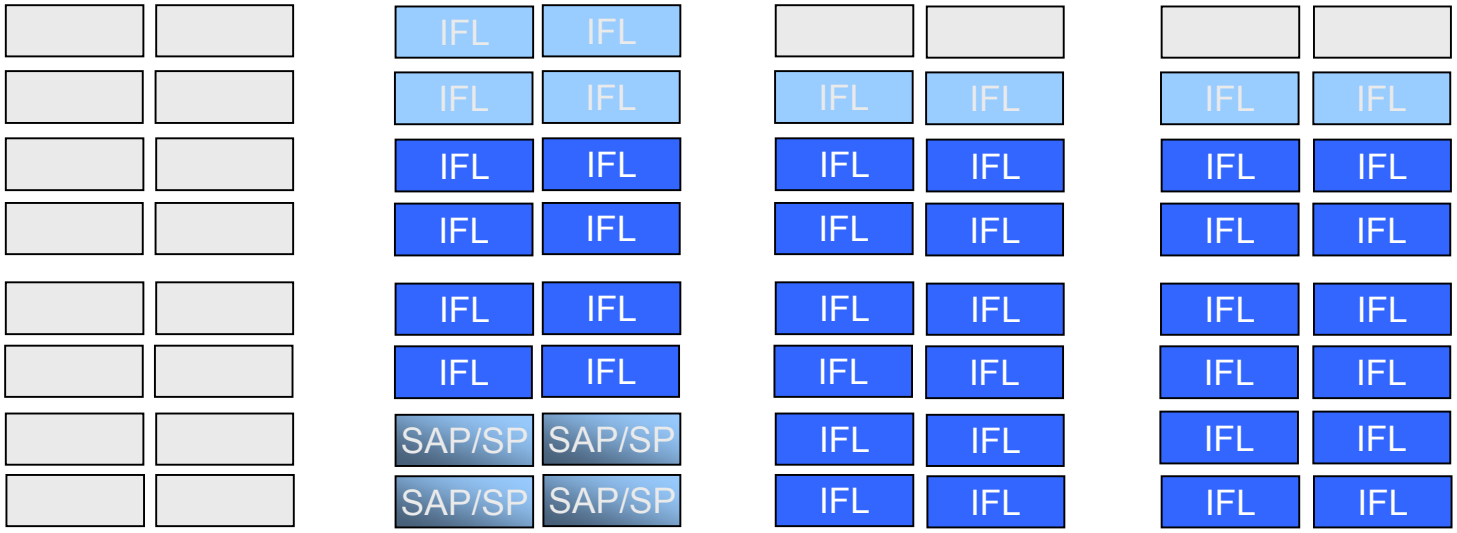
Network
22 OSA Ports
Max 48 CHPIDs
(45% *)



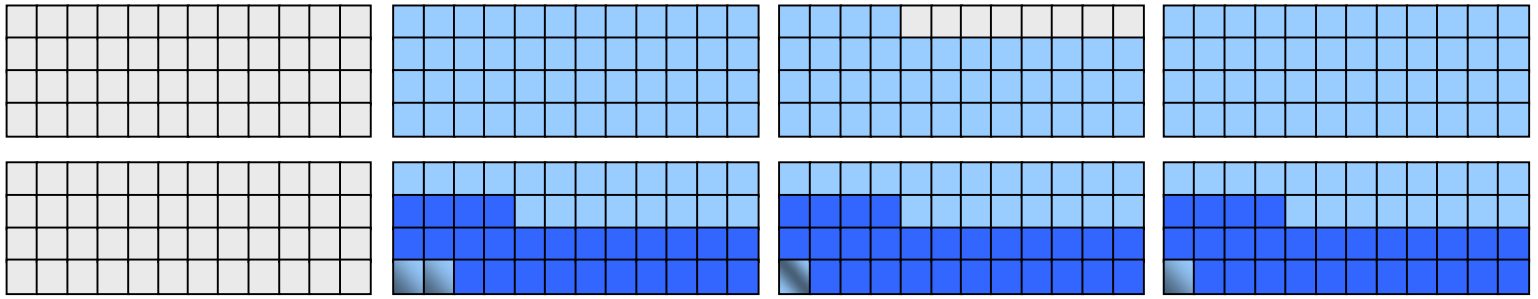
SHARE in Boston

IBM z10 Platform (Prod)

Processors
33 IFLs
Max 40 (3 books)
(83%)
Max 64 (4 books)
(52%)



Memory
352GB
Max 1136GB
(31%)
Max 1520GB
(23%)

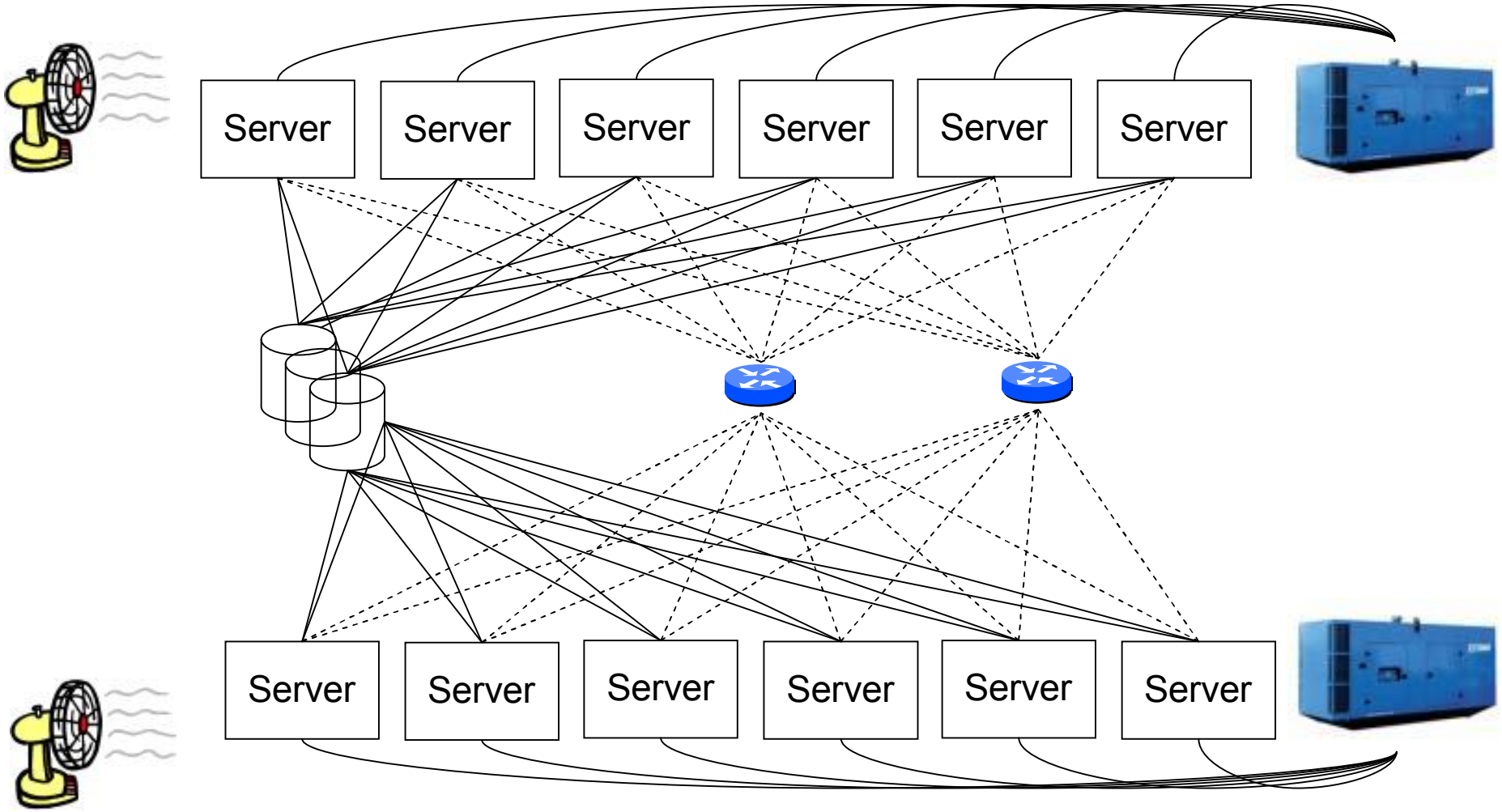


Network
10 OSA Ports
Max 48 CHPIDs
(21% *)

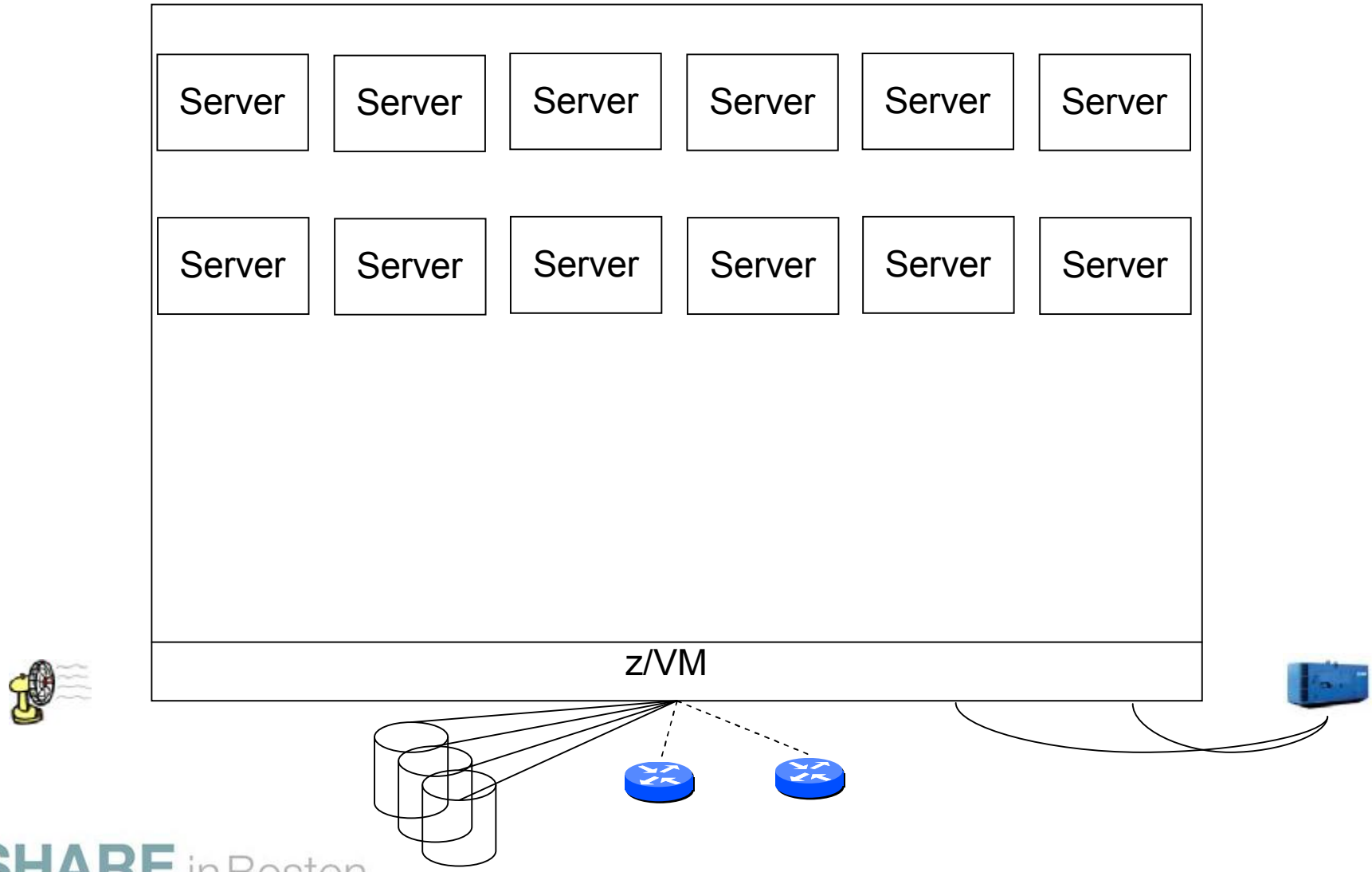
Simple “Logical” TCO
Why Virtualization on System z?

- Reduce complexity
  - Physical servers
  - Network connections
  - Disk connections
- Reduce facility resources
  - Floor space
  - Power consumption
  - Cooling
- Opportunities
  - Shared disk
  - Shared memory
  - Reduced total capacity because of sharing

Distributed Server Model



Virtual Server Model



Why Virtualization on System z?

- 19” Server Rack with 1U x86 (e.g. HP Proliant DL360 G4)
  - Physical width 20” per rack
  - 40 servers per rack
  - Max servers in 3 racks is 120
  - 585W * 40 * 3 = 70.2kW
  - 2kBTU/hr * 40 * 3 = 240kBTU/hr
  - 120 servers

Note: The numbers on this and the following charts are approximate and are based on information located on vendor sites on the Internet.

Why Virtualization on System z?

- 19” Server Rack with 4U x86 (e.g. HP ProLiant DL580 G5)
  - Standard Rack: physical width 20” per rack; 42 rack units
  - 10 servers per rack
  - Max servers in 3 racks is 30
  - Power: 964W (per power supply) * 2 * 10 * 3 = 57.8kW
  - Heat: 3355BTU/hr * 10 * 3 = 100kBTU/hr
  - Sockets: 2 * 10 * 3 = 60 Sockets (4 sockets available)
  - 4-Core Xeon: 60 * 4 = 240 Cores (6-Core processors available)
  - 60 servers
    VMware will permit more virtual servers – how many depends on workload
    - Theoretical VMware maximum 8 * 240 = 1920 virtual servers
    - Realistic maximum 20 (or less) * 30 = 600 virtual servers

Why Virtualization on System z?

- z10 EC – Model 2097-E40
  - Physical width 60”; about 3 19” racks
  - Power:
    - 9.70kW E12 1 book, 1 I/O cage (Model E16)
    - 27.50kW E64 4 books, 3 I/O cages (Model E64)
    - 24.40kW E40 3 books, 3 I/O cages & fully populated with I/O cards
  - Heat:
    - 32.98kBTU/hr E12 1 book, 1 I/O cage (Model E16)
    - 93.50kBTU/hr E64 4 books, 3 I/O cages (Model E64)
    - 82.96kBTU/hr E40 3 books, 3 I/O cages & fully populated with I/O cards

Why Virtualization on System z?

- z10 EC – Model 2097-E64
  - Development/Test
    - 391 servers using 33% of IFLs and 28% of memory
      IHS 91, WAS 194, DB2 64, Tools 28, Other 14
      JVM / IHS instances 1786
    - 17.853 kW (kVA); 60.9kBTU/hr @ 55% utilization
      18.025 kW (kVA); 61.5kBTU/hr @ 94% utilization
      17.972 kW (kVA); 61.3kBTU/hr @ 98.6% utilization
    - z10 maximum:
      - 1184 servers for production (391/33% of IFLs currently used)
        5412 JVM / IHS instances
  - Production
    - 196 servers using 50% of IFLs and 21% of memory
      IHS 53, WAS 102, DB2 34, Tools 7
      JVM / IHS instances 911
    - 14.749 kW (kVA); 50325 BTU/hr @ 33% utilization
    - z10 maximum:
      - 377 servers for production (196/52% of IFLs currently used)
        1752 JVM / IHS instances

Why Virtualization on System z?

- 19” Server Rack with 1U x86 (e.g. HP Proliant DL360 G4)
  - Software (25% web server; 55% application server; 20% database server)
    Assumption: 2/3 of servers are at least dual processor

    To run 120 servers with 120 applications
    - 180 Core licenses for OS
    - 30 web server licenses (could choose to use “free” open source)
    - 132 WAS licenses – 55% of 60 * 2 Cores
    - 48 DB2 or Oracle licenses (no DB servers with only one processor)
    - Other software licenses? (e.g. MQ)

Note:

On x86, software is licensed per processor core. If these 120 servers are dual or quad core, the costs double or quadruple.

For the DL580 hardware above, software licenses would be paid on 240 cores.

Why Virtualization on System z?

- z10 EC – Model 2097-E64
  - Software (25% web server; 55% application server; 20% database server)
    To run 380 servers with 2000+ applications
    - 64 licenses for OS
    - 64 web server licenses (could choose to use “free” open source)
    - 64 WAS licenses
    - 64 DB2 or Oracle licenses
    - Other software licenses? (e.g. MQ)

Why Virtualization on System z?

- z10 to x86 1U DL380 G4 comparison
  - 314% servers – 377 vs 120
  - 1460% applications – 1752 vs 120
  - 53% OS licenses – 64 vs 120
    48% WAS licenses – 64 vs 132
    130% DB2 licenses – 64 vs 48
  - 40% power – 28kW vs 70kW
  - 39% cooling – 94kBTU vs 240kBTU
- z10 to x86 4U DL580 G5 comparison
  - 63% servers – 377 vs 600
  - 299% applications – 1792 vs 600
  - 27% OS licenses – 64 vs 240
    48% WAS licenses – 64 vs 132 (requires segregation of workloads)
    130% DB2 licenses – 64 vs 48 (requires segregation of workloads)
  - 48% power – 28kW vs 60kW
  - 94% cooling – 94kBTU vs 100kBTU

Why Virtualization on System z?

- 19” Server Rack with 1U x86 (e.g. HP Proliant DL360 G4)
  - Cabling (25% web server; 55% application server; 20% database server)
    - Many GB of dedicated disk – frequently underutilized
    - 91-182 SAN connections (91 servers with SAN connection; dual path)
      Assumption: All DB and half Application servers use SAN disk
    - 240+ Ethernet connections (2 per server minimum; most have 3)
    - Serial connections for console most likely

Why Virtualization on System z?

- z10 EC – Model 2097-E40
  - Cabling (25% web server; 55% application server; 20% database server)
    - 4 4Gb FICON ECKD DASD (Disk) connections
    - 12 4Gb FCP SAN connections
    - 10 1Gb Ethernet connections

Why Virtualization on System z?

- 19” Server Racks with 1U x86
  - 450-900 servers
    450-900 applications
  - 120+ software licenses
  - 240+ Gb Ethernet connections
  - 100(?) SAN connections
  - 70kW
  - 240kBTU/hr

- z10 EC – model 2097-E40
  - 375-1180 servers
    1750-5400 applications
  - 64 software licenses
  - 10 Gb Ethernet connections
  - 12 SAN connections
  - 4 Disk connections
  - 18kW
  - 60kBTU/hr

  - **2-4x applications**
  - **1/4x power and cooling**

Virtual Networking

- Overcoming Terminology – for the network novice
  - VLAN, VLAN, Guest LAN
    - VLAN – native, hardware, management – the one the routers, switches and OSAs use
    - VLAN – logical – the ones used to separate/isolate servers
    - Guest LAN – a VM emulation of a network
  - Switches, Routers, VSWITCH
    - Switch – a device that acts as a connector to create a network
    - Router – a device that forwards data packets between computer networks
    - VSWITCH – a logical extension of the physical network inside the System z

Virtual Networking

- System z Hardware
  - Open System Adapter (OSA) Express 2
    - 2 CHPIDs / 2 Ports per card
    - Gigabit adapter with a smart network controller
    - Two types
      - Gigabit Ethernet
        - Fiber
      - 1000BaseT
        - Copper Cat6
        - Can be configured as Integrated Console Controller (ICC)
  - Open System Adapter (OSA) Express 3
    - 2 CHPIDs / 4 Ports per card
  - System z 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

Virtual Networking

- System z Hardware with z/VM
  - Virtual Switch (VSWITCH)
    - Combination of System z microcode and z/VM CP code to create an extension of a network switch
    - Layer 3
      - Forwarding based on IP address
      - Sufficient for most implementations
      - Defined as "IP"
      - Common MAC included for all guests
      - z/VM 4.4.0 or higher

Virtual Networking

- System z Hardware with z/VM
  - Virtual Switch (VSWITCH)
    - Layer 2
      - Forwarding based on MAC address
      - Allows non-IP protocols like NETBIOS or IPX
      - Defined as "Ethernet"
      - New on z990 with OSA Express 2 and z/VM 5.1.0
      - Recommended by IBM
      - Unique MAC for each virtual server
        - Local MAC addressing must be administered
      - z/VM TCPIP can connect to a Layer 2 VSWITCH as of z/VM 5.4.0
      - Useful for newer functions such as Link Aggregation

Virtual Networking

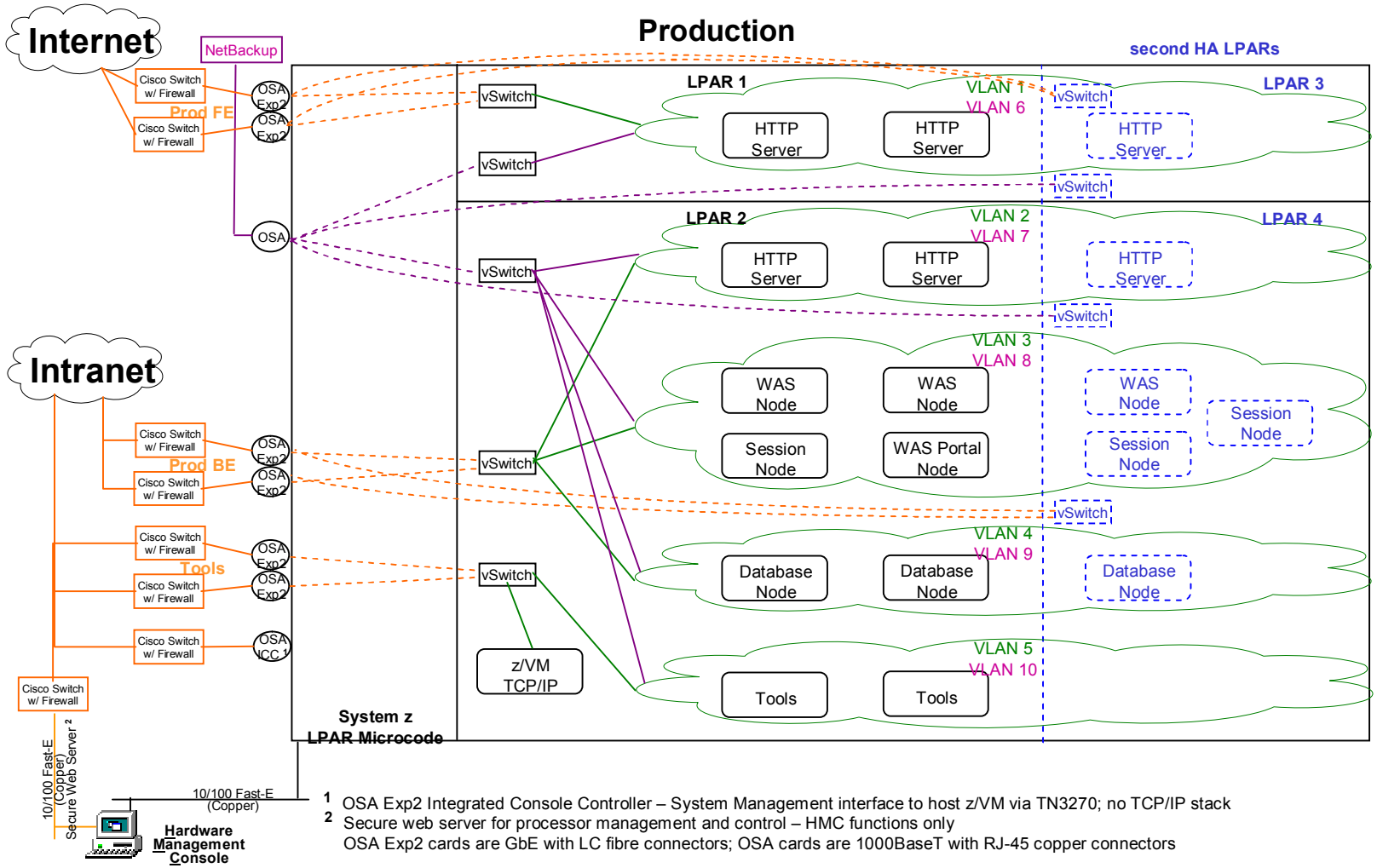
- z/VM
  - Guest LAN
    - Use to create isolated LAN within a z/VM LPAR
      - Can be owned by SYSTEM or a virtual machine
      - Can be restricted to authorized users or open to anyone
    - HIPERSOCKET – emulate System z HIPERSOCKET hardware
    - QDIO – emulate gigabit Ethernet
      - IP – Layer 3 level of the OSA model
      - ETHERNET – Layer 2 level of the OSI model
    - Define in CP SYSTEM CONFIG or by CP command

```
DEFINE LAN GLAN1 OWNERID SYSTEM TYPE HIPERS MAXCONN INFINITE
DEFINE LAN GLAN2 OWNERID SYSTEM TYPE QDIO IP MAXCONN INFINITE
DEFINE LAN GLAN3 OWNERID SYSTEM TYPE QDIO ETHERNET MAXCONN INFINITE
```

Our Network

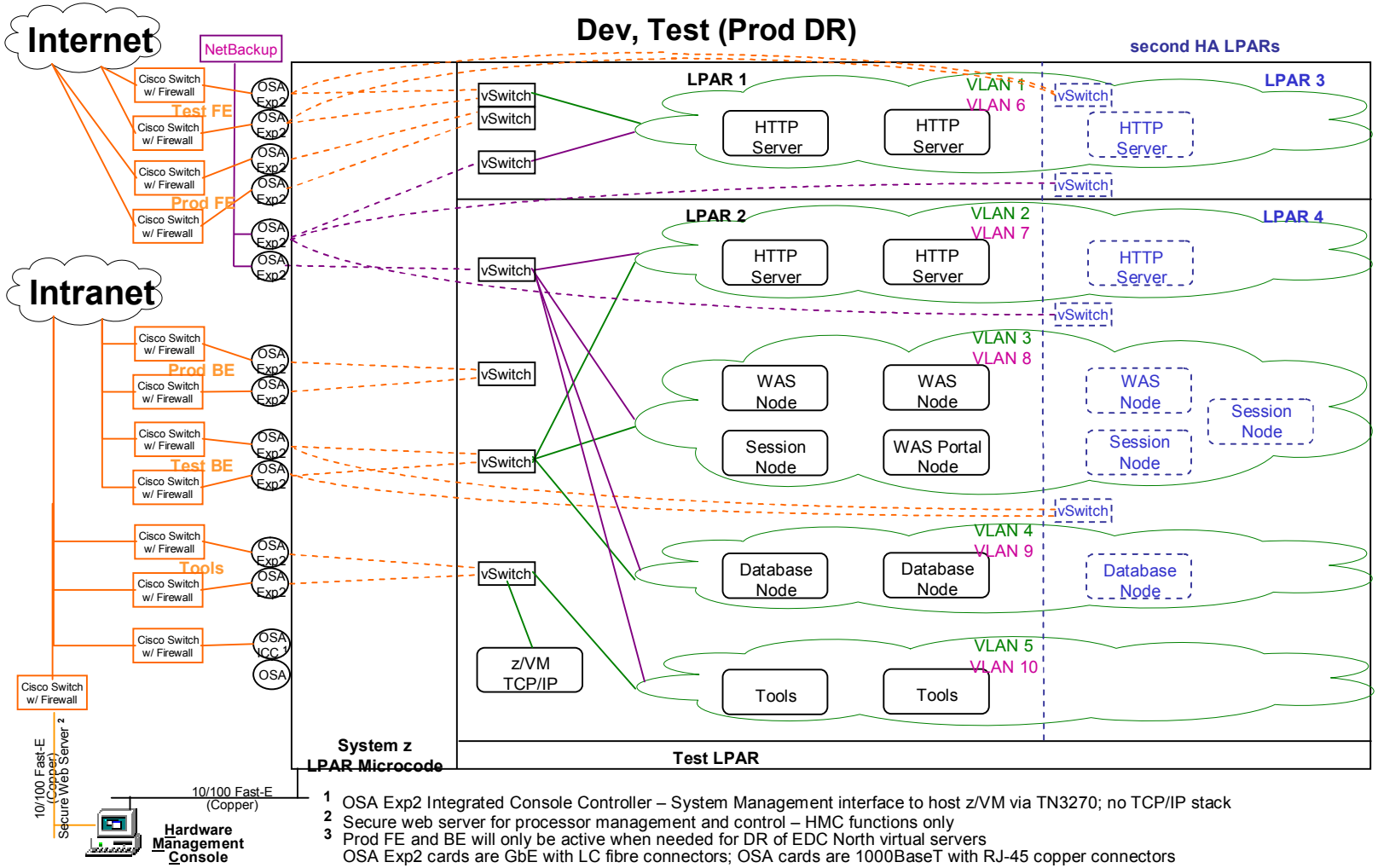
- Our OSA / VSWITCH configuration
  - Production
    - 3 OSA Express 3 Gigabit Ethernet cards
      12 ports; 8 used
    - 2 OSA Express 2 1000BaseT Ethernet cards
      4 ports; 3 used
  - Development/test
    - 8 OSA Express 3 Gigabit Ethernet cards
      32 ports; 16 used
    - 2 OSA Express 2 1000BaseT Ethernet cards
      4 ports; 2 used
  - 9 different network zones; 15 VSWITCHes defined
    - 2 VSWITCHes on each pair of OSA ports for redundancy and load distribution
      - Paired OSA ports are on separate cards for redundancy
    - Each pair of ports is in a specific network zone
      - Each OSA port in a pair is connected to a different physical switch

Our Network



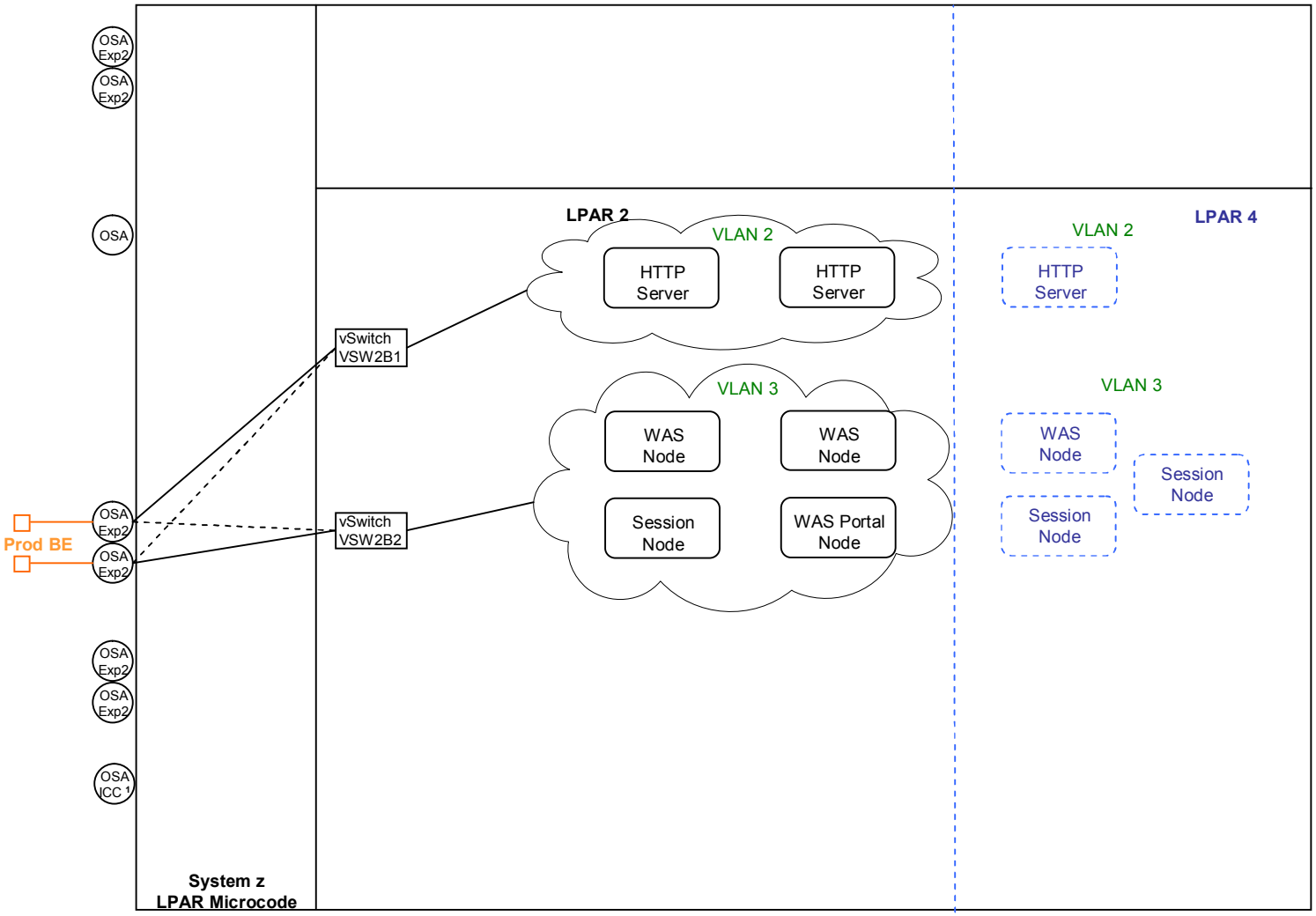
1 OSA Exp2 Integrated Console Controller – System Management interface to host z/VM via TN3270; no TCP/IP stack
2 Secure web server for processor management and control – HMC functions only
OSA Exp2 cards are GbE with LC fibre connectors; OSA cards are 1000BaseT with RJ-45 copper connectors

Our Network

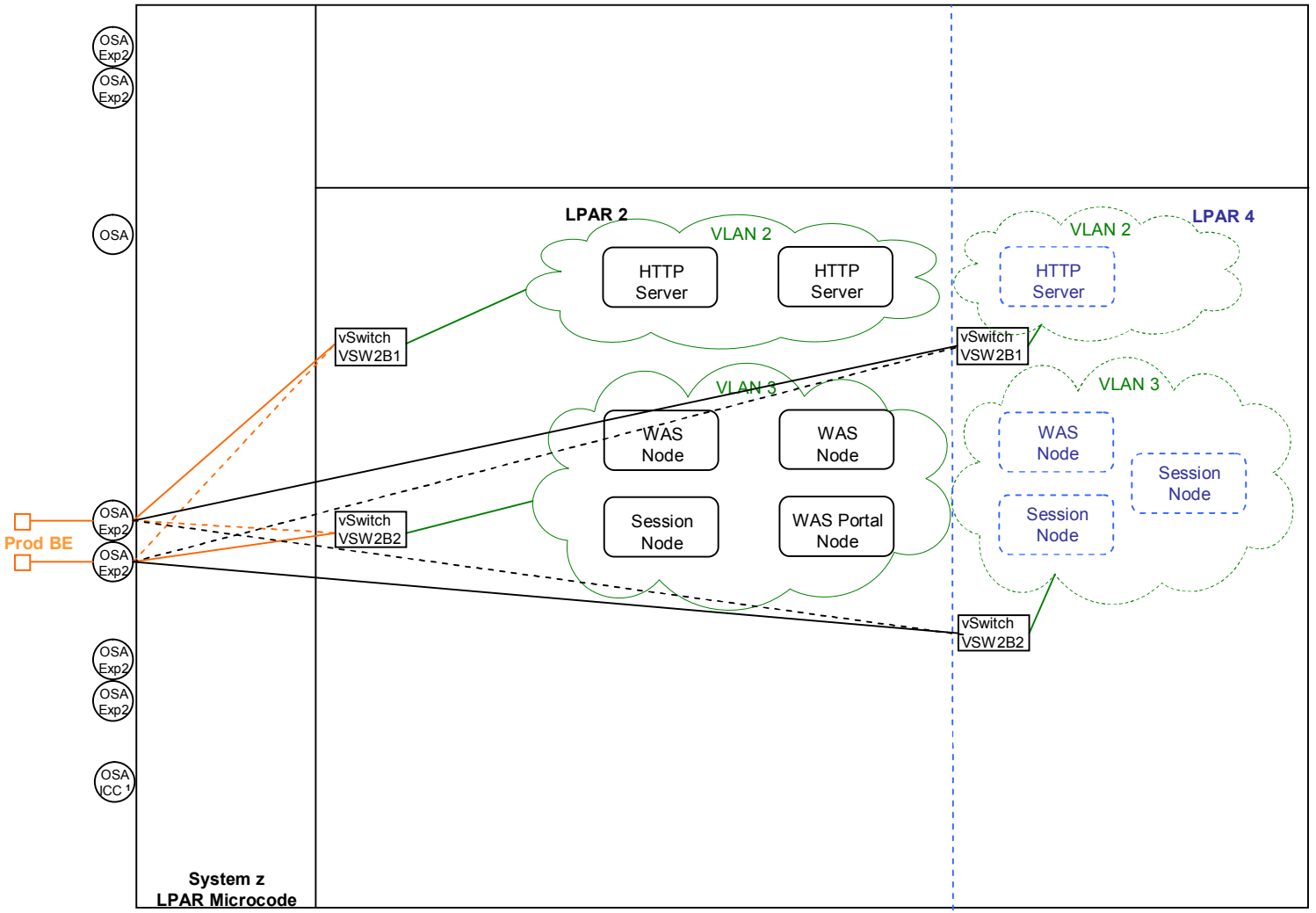


1 OSA Exp2 Integrated Console Controller – System Management interface to host z/VM via TN3270; no TCP/IP stack
2 Secure web server for processor management and control – HMC functions only
3 Prod FE and BE will only be active when needed for DR of EDC North virtual servers
OSA Exp2 cards are GbE with LC fibre connectors; OSA cards are 1000BaseT with RJ-45 copper connectors

Network



Network



VSWITCH Detail

- Defining VSWITCH
  - In SYSTEM CONFIG or via CP command by authorized user (same syntax in both places)
    - Example of a pair of VSWITCHes:

```
CP DEFINE VSWITCH VSW2B1 RDEV C100 C204 CONTROLLER * IP VLAN 4094
CP DEFINE VSWITCH VSW2B2 RDEV C200 C104 CONTROLLER * IP VLAN 4094
```

    - VLAN on the VSWITCH is the default VLAN used by the **hardware** switches.

VSWITCH Detail

- Authorizing virtual servers to use VSWITCH
  - SYSTEM CONFIG format
    - Example 1: 2 virtual servers in same zone on opposite VSWITCHes

```
MODIFY VSWITCH VSW2B1 GRANT LINSERV1 VLAN 1001
MODIFY VSWITCH VSW2B2 GRANT LINSERV2 VLAN 1001
```

    - Example 2: 1 virtual server on 2 VSWITCHes in different zones

```
MODIFY VSWITCH VSW2B1 GRANT LINSERV1 VLAN 1001
MODIFY VSWITCH VSW2F1 GRANT LINSERV1 VLAN 2001
```

  - CP command format
    - Example 1: 2 virtual servers in same zone on opposite VSWITCHes

```
CP SET VSWITCH VSW2B1 GRANT LINSERV1 VLAN 1001
CP SET VSWITCH VSW2B2 GRANT LINSERV2 VLAN 1001
```

    - Example 2: 1 virtual server on 2 VSWITCHes in different zones

```
CP SET VSWITCH VSW2B1 GRANT LINSERV1 VLAN 1001
CP SET VSWITCH VSW2F1 GRANT LINSERV1 VLAN 2001
```

VSWITCH Detail

- Defining Guest Virtual NIC
  - CP DIRECTORY format

```
NICDEF 5708 TYPE QDIO DEVICES 3 LAN SYSTEM TOOL2
NICDEF 1E00 TYPE QDIO DEVICES 3 LAN SYSTEM NETBKUP1
```

  - CP command format

```
CP DEFINE NIC 5708 TYPE QDIO DEVICES 3
CP COUPLE 5708 TO SYSTEM TOOL2
CP DEFINE NIC 1E00 TYPE QDIO DEVICES 3
CP COUPLE 1E00 TO SYSTEM NETBKUP1
```

VSWITCH Detail – Linux definitions

- Hardware configuration script

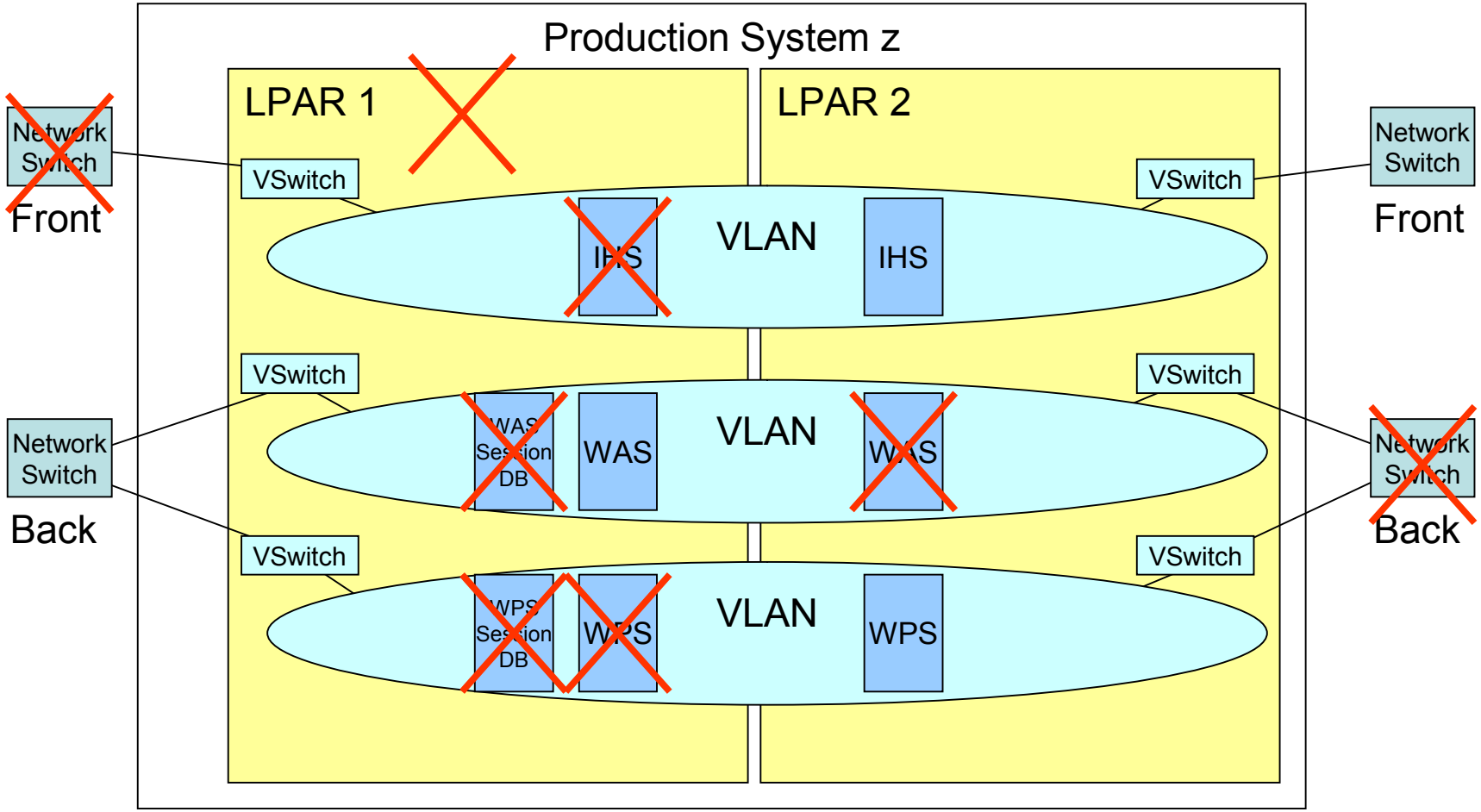
```
cat /etc/sysconfig/hardware/hwcfg-qeth-bus-ccw-0.0.5708
#!/bin/sh
#
# hwcfg-qeth-bus-ccw-0.0.5708
#
# Hardware configuration for a qeth device at 0.0.5708
# Automatically generated by netsetup
#
STARTMODE="auto"
MODULE="qeth"
MODULE_OPTIONS=""
MODULE_UNLOAD="yes"
# Scripts to be called for the various events.
SCRIPTUP="hwup-ccw"
SCRIPTUP_ccw="hwup-ccw"
SCRIPTUP_ccwgroup="hwup-qeth"
SCRIPTDOWN="hwdown-ccw"
# CCW_CHAN_IDS sets the channel IDs for this device
# The first ID will be used as the group ID
CCW_CHAN_IDS="0.0.5708 0.0.5709 0.0.570a"
# CCW_CHAN_NUM set the number of channels for this device
# Always 3 for an qeth device
CCW_CHAN_NUM=3
# CCW_CHAN_MODE sets the port name for an OSA-Express device
CCW_CHAN_MODE="suselin7“
# QETH_LAYER2_SUPPORT: 1 for Layer 2, 0 for Layer 3
QETH_LAYER2_SUPPORT=0
```

VSWITCH Detail – Linux definitions

- Confirmation

```
ifconfig
eth0      Link encap:Ethernet  HWaddr 02:00:00:00:00:05
          inet addr:10.1.1.1  Bcast:10.1.1.1  Mask:255.255.255.0
          inet6 addr: fe80::200:0:100:5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:652 (652.0 b)
eth1      Link encap:Ethernet  HWaddr 02:00:00:00:00:04
          inet addr:10.2.1.1  Bcast:10.2.1.1  Mask:255.255.255.0
          inet6 addr: fe80::200:0:100:4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:150122 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66742 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:32348101 (30.8 Mb)  TX bytes:17319537 (16.5 Mb)
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

High Availability
Failure Scenarios Tested



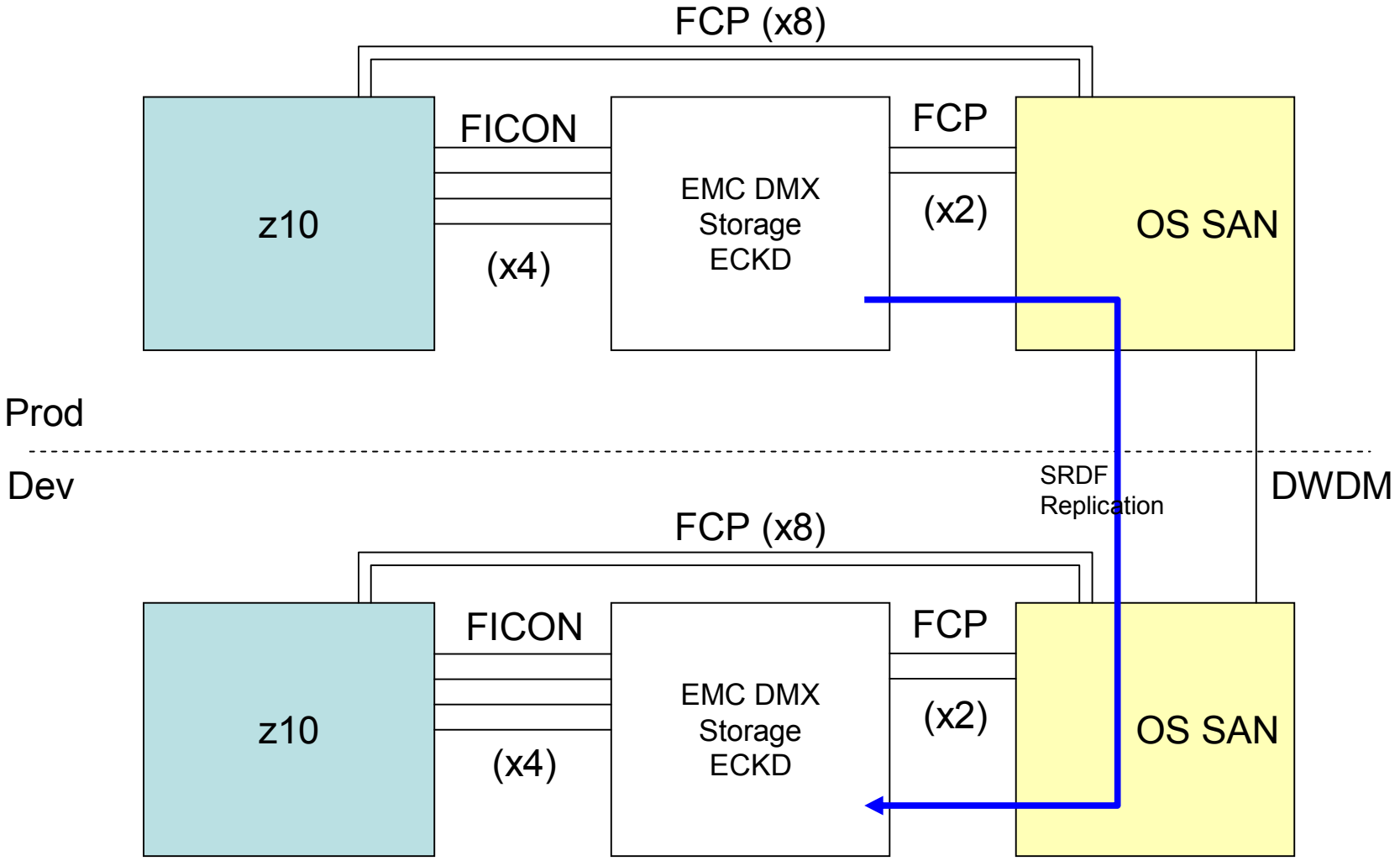
High Availability Clustering

- Scenarios tested
  - Loss of clustered web server
  - Loss of network switch
  - Loss of clustered application server
  - Loss of entire z/VM LPAR
- Current Limitations
  - Single System z
    Increased availability if LPARs are spread across CPCs

Disaster Recovery

- Included with our Linux on System z offering
  - Disk replication between sites
  - Complete server configuration stored at second site
  - Physical network connections in place
  - Standby network definitions
  - Automated script for network personality at second site
    - Script on virtual server "asks" where it is running and sets network parameters
    - External DNS swap process must be performed
  - If primary site is unavailable, virtual servers are defined and booted at second site

Disaster Recovery - Disk



Performance Measurement Tools

Performance Basics (very basic)

- Basic metrics to watch – z/VM
  - CPU utilization
    - While System z runs fine at 100%, Linux workload is much more demanding than traditional mainframe workloads.
      - Significant impact of memory over-commit
      - May need to keep peak periods at 85-90%.
  - Memory
    - Many Linux guests have huge working set sizes and many don't go idle
    - Keep memory over-commit less than 2:1
      (ratio of combined working set sizes to real memory available)
  - Paging
    - z/VM has no problem with high page rates
      - Keep Expanded Storage for high-speed page buffer
    - Guests may not be tolerant
    - Allocate enough VM page space for twice the total of the working set of expected guests; Use CP QUERY ALLOC PAGE to monitor and keep page space less than 50% full

Performance Basics

- Basic metrics to watch – Linux Guests
  - Don't wake guests to ask
    - Choose performance tools that understand that Linux is running on z/VM
  - Pick **one** tool
    - Multiple monitoring tools adds a lot of overhead
    - ½% CPU per server adds up fast when there are 100s of servers
  - CPU measured inside guest is not very meaningful
    - Improved with CPU accounting enhancement at later kernel levels SLES10 and RHEL4
  - Avoid TOP – significant overhead
    - Use vmstat or nmon

Performance Basics

- Basic metrics to watch – Linux Guests
  - Memory
    - Don't over provision. Large virtual storage sizes drive up z/VM paging.
    - Use a swap hierarchy with z/VM VDISK as the highest priority swap space. It is not a problem for Linux to do some swapping.
    - Show a snapshot of memory/swap: free or cat /proc/meminfo
    - Avoid multiple I/O caching in DB2
      - Tell DB2 to read directly from disk to buffer pools; rely on z/VM I/O caching
        - Pre-8.2 directio=yes
        - No File System Caching for 8.2 and later
    - Default Linux memory management may not be optimal
      - Kernel parm: vm.swapiness=60
        Default may be too high – causes memory to be consumed
        Lower values cause Linux to reuse memory allocations more often to reduce memory demand
  - Paging
    - Prevent Linux from paging. z/VM paging is much more efficient.
    - Show Linux pagein/pageout: cat /proc/vmstat | grep pgpg

Performance Basics

- Basic metrics to watch – Linux Guests
  - Look at guest CPU demand from z/VM
  - Watch for excessive paging on behalf of a guest.
    - May indicate inefficient memory usage or excessive virtual storage allocation
  - Watch for guests with poor I/O response
    - System z handles high I/O rates fine but bottlenecks can occur
  - Watch for % of active time that guests spend in various queues
    - Run
    - CPU queue
    - Page queue
    - etc

Performance Basics

- Linux Guests internal performance
  - Tools to analyze guests functions vary greatly
    - Some have a lot of tools – WAS
    - Some may have little to offer
  - Application developers debugging skills may be limited
    - Accustomed to working with excessive capacity
    - Not accustomed to shared environment

Performance Basics

- Ideas that may help
  - Utilize Cryptographic hardware
    - Dramatically improves SSL calls for secure web pages
  - Minimize external network hops
    - Use virtual firewall solutions
    - Staying inside the System z hardware operates at memory speeds
  - Reduce NTP frequency
  - Minimize or stagger cron scheduling

Performance Future Options

- Shared Read-Only disks
  - Requires separation of code from configurations
    - Special mount points or symbolic links
  - Default for all provisioned guests
- Cooperative Memory Management
  - Some testing done; working on automated control; implementation pending
- Execute In Place (xipfs)
- DCSS – shared code in z/VM storage

Conclusions

- Linux virtualization on System z does:
  - Reduce complexity
  - Improve provisioning time
    - No hardware acquisition
    - No physical installation to perform
  - Reduce environmental demand
    - Less cooling
    - Less power
    - Less floor space
  - Easy technology upgrades

Conclusions

- Beware of outdated information
  - Technology is changing very quickly
  - Periodic repetition of comparisons is necessary
- Carefully evaluate any published comparisons
  - Apply a generous dose of reality
  - Avoid theoretical data
  - Factor in results of controlled testing of real applications

Conclusions

- Things are changing rapidly
- Prepare for availability and continuity
- Performance is an iterative learning process
  - Difficult to understand all available data without significant immersion
- Be careful what you ask for because you may get it!

Contact Information

"And I thought we were busy ***before*** Linux showed up!"

Rick Barlow
Senior z/VM Systems Programmer

Phone: (614) 249-5213

Internet: Richard.Barlow@nationwide.com