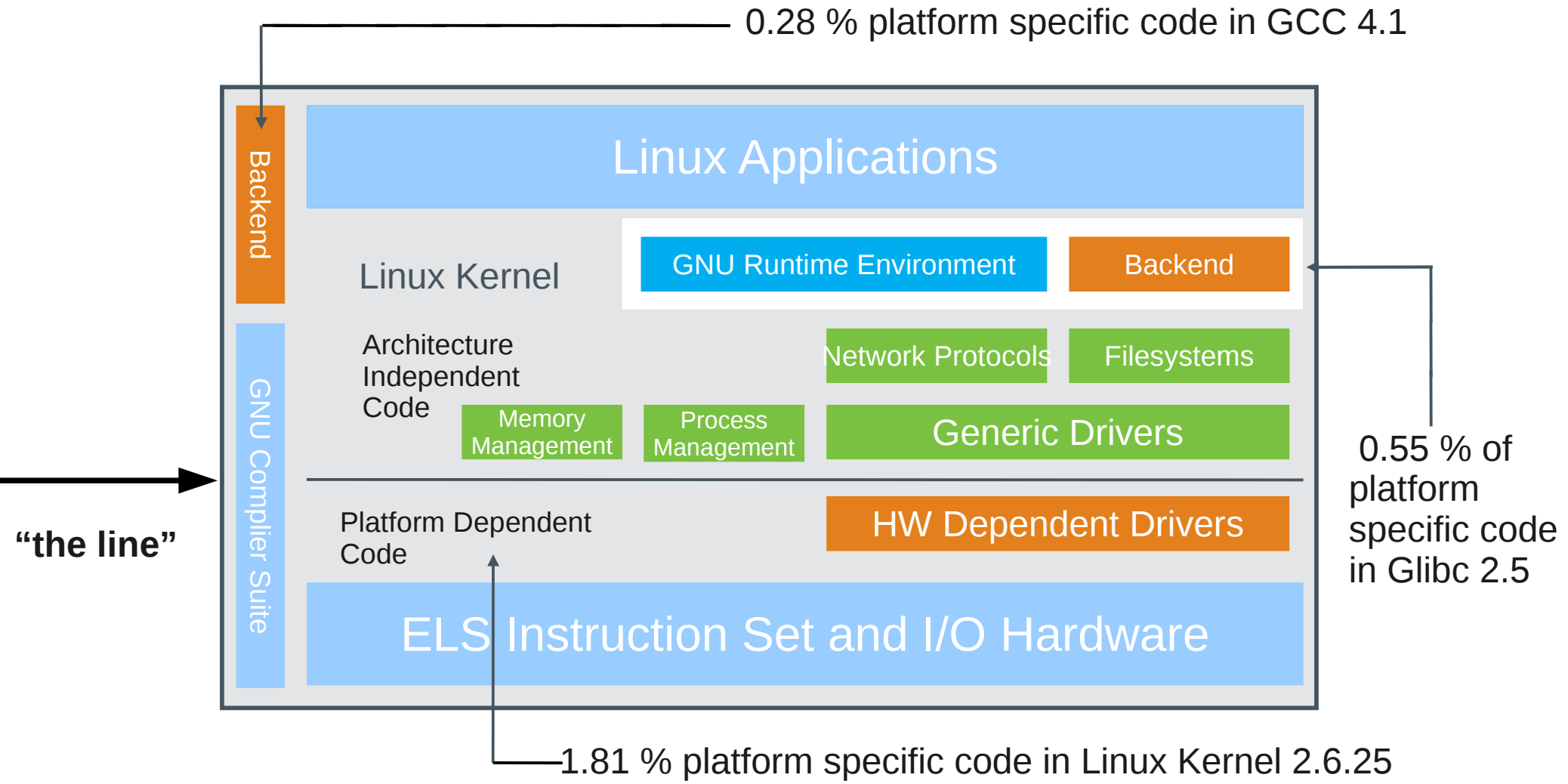
Current & Future State of Red Hat Enterprise Linux on System z

Brad Hinson
Worldwide Lead, Linux on System z
bhinson@redhat.com

redhat

Structure of Linux on System z

Many Linux software packages did not require any code change to run on Linux on System z



RHEL 6 on System z, “Above the Line”

RHEL 6: Key Features

- **Power Management**
  - Improvements through the application stack to reduce wake ups
  - Power consumption measurement by Powertop
  - Power Management and adaptive system tuning by Tuned
- **Next Generation Networking**
  - Comprehensive IPv6 support (NFS 4, CIFS, ISATAP support)
  - Redesigned QETH network driver with support for OSA data connection isolation
- **Reliability, Availability, and Serviceability**
  - System level enhancements from industry collaborations to make the most of hardware RAS capabilities.

RHEL 6: Key Features

- **Fine-grained Control and Management**
  - Improved scheduler and better resource management in the kernel via Completely Fair Scheduler (CFS) and Control Groups (CG).
- **Scalable File systems**
  - New file systems btrfs and ext4 (now the default) offer robustness, scalability, and high-performance.
- **Disk Storage Subsystem**
  - FCP automated port discovery and lsluns utility to automatically activate all available target ports
  - Support for High Performance FICON to reduce I/O overhead
  - Dynamically adjustable queue depth
  - I/O configuration support when running in LPAR

RHEL 6: Key Features

- **Virtualization**
  - Tighter integration with z/VM for extended functionality like dynamic memory resizing, better CPU utilization, HyperPAV, and suspend/resume support.
- **Enterprise Security Enhancement**
  - SELinux includes improved ease of use, application sandboxing, and significantly increased coverage of system services
  - SSSD provides unified access to identity and authentication services as well as caching for off-line use.
  - Support for the latest Crypto Express3 accelerator and coprocessor for offloading the processing of secure data.
- **Development and Runtime Support**
  - SystemTap (allows instrumentation of a running kernel without recompilation) and ABRT (simple collection of bug information)
  - Improvements to glibc (version 2.12), GDB (version 7.1), and the GCC compiler (version 4.4), which can lead to greater than 10% performance improvement.

Efficiency, Reliability & Scalability

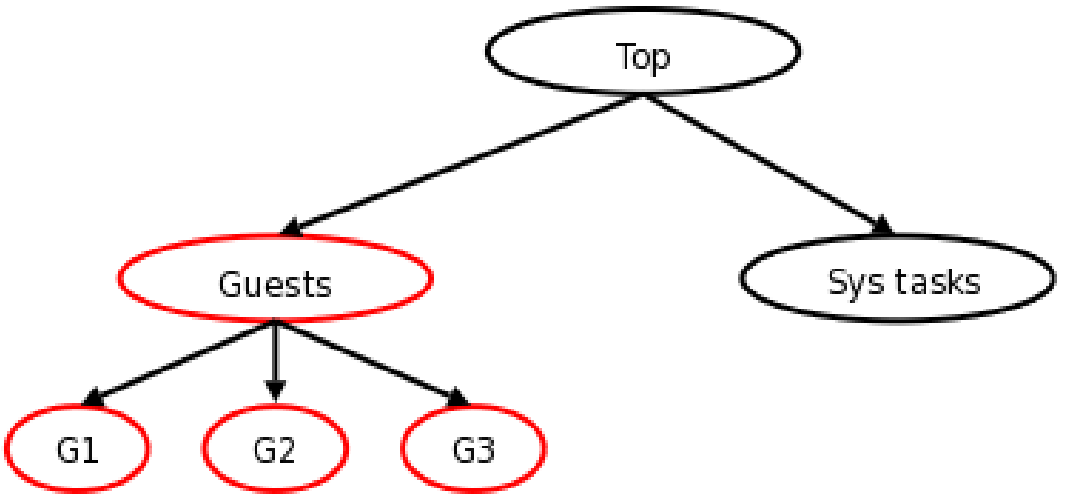
- Efficient Scheduling
  - CFS (Completely Fair Scheduler) is a new application scheduler which operates with nanosecond granularity
  - Optimized for multi-core topologies, CFS maximizes CPU utilization with low power consumption
- Reliability
  - With support for processor degradation and other machine checks, the system can recover from fatal hardware errors with minimal disruption
  - Memory pages with errors can be declared as “poisoned” and will be avoided
- Scalability
  - Full support for up to 80 IFLs and up to 3 TB memory on z196
  - Support for up to 4 million processes per server

In Depth: Cloud Enablement with Control Groups

Problem: “I want to implement a chargeback model.”

Solution: Congrol Groups (cgroups)

- Cgroups are “process containers”. Lets you transform groups of applications into workloads



redhat

In Depth: Cloud Enablement with Control Groups

- **Resource Limiting**

  Specify limits on CPU, memory, and even file system usage

- **Prioritization**

  Give mission critical workloads higher priority than others

- **Accounting**

  Run report on resource utilization, i.e. for billing purposes

- **Isolation**

  Separate namespaces for groups, so they don't see each other's processes, network connections or files

- **Control**

  Freeze groups for checkpointing or restarting workloads

Filesystems

- Ext4 replaces Ext3 as default filesystem
  - Faster, more robust, and scales to 16 TB
  - Support for thin provisioning. You can create a 16 TB file system on a single mod-3 DASD! Just add more when needed.
- NFS version 4 for network file system
  - Clustered file system with support for read/write access from multiple guests simultaneously.
  - Use VSWITCH for fast “network” access, or Hipersockets for memory-speed transfers.
- Fuse
  - Allows filesystems to run in user space, allowing testing and development of fused-based filesystems.
  - For example, use Fuse to create a cloud filesystem.

redhat

In Depth: Next Generation BTRFS Filesystem

- Provides enhanced data integrity, better performance, and ease of use over Ext3/Ext4 filesystems.
- Key Features
  - Copy-on-write for cloning and snapshots, with filesystem seeding
    - Create a filesystem to seed other BTRFS filesystems. The original filesystem is a read-only starting point, the copy-on-write ensures the original is unchanged.
  - Online defragmentation
  - Load balancing across device nodes
  - Transparent data compression
  - In-place conversion (with rollback) from Ext3/Ext4
  - Data deduplication (under development)
- Principle developer for Ext3/Ext4 stated BTRFS is the way forward, having a number of the same design ideas that Reiser had.

LVM, Storage & Multipath

- LVM snapshot improvements

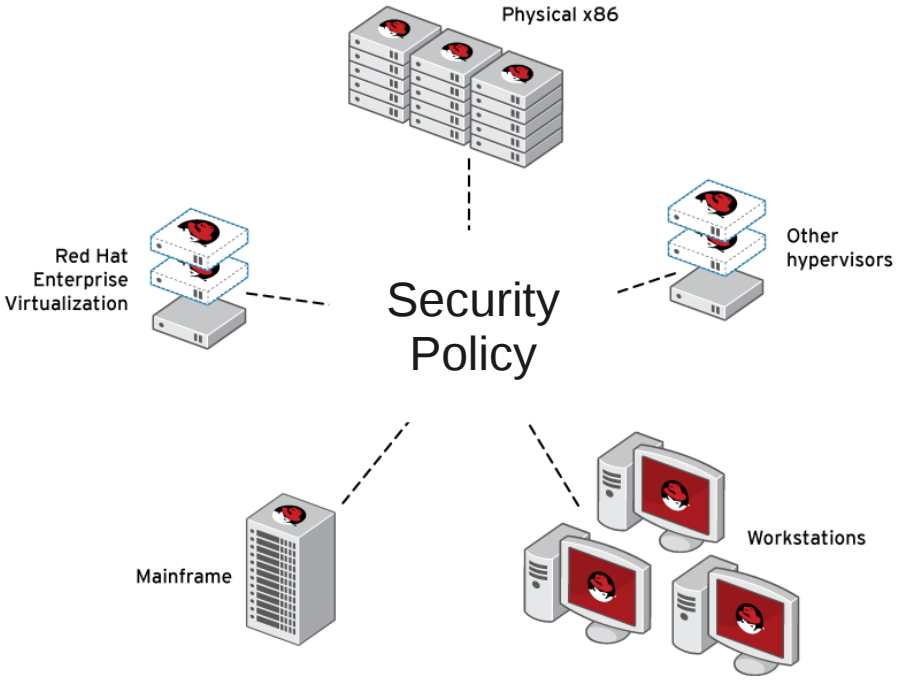
  - Snapshot can now be merged back into the original logical volume, reverting changes that occurred after the snapshot.
  - Practical application: “lock” your data with regular snapshots, then if necessary “undo” any changes within the last week/day/hour..

- Define LVM hot spare
  - Recover seamlessly from device failure of a disk or logical volume
- Device Mapper Multipath
  - Allows paths to be dynamically selected based on queue size or I/O time data
- Performance and Integrity
  - Automated I/O alignment and self-tuning
  - DIF/DIX provides better integrity checks for application data

- UDP Lite tolerates partially corrupted packets to provide better service for multimedia protocols, such as VOIP, where partial packets are better than none.
- Multiqueue Networking increases processing parallelism for better performance from multiple processors and CPU cores.
- Large Receive Offload (LRO) and Generic Receive Offload (GRO) aggregate packets for better performance.
- Support for datacenter bridging includes data traffic priorities and flow control for increased quality of service.
- iSCSI partitions may be used as either root or boot filesystems.

IPA Client (Identity, Policy, Audit)

- **IPA Client in Core RHEL for Centralized Security Management**
  - Kerberos authentication with host based access control
  - Provides central storage of extended user attributes
  - Enables centralized policy for applications, including SELinux policy
  - Audit log aggregation services & search capabilities



Security enhancements

- **Virtualization Isolation in Conjunction with SELinux**
  - Labeled NFS for filesystem isolation, guest confinement via SELinux policy enhancements
- **NSS Crypto**
  - Broaden the core services which utilize NSS crypto libraries
  - Allows cheaper implementation of new features, ie TPM & centralized key store
  - Incremental targeted conversion of: Openswan, openldap, glibc
  - Add new crypto GUI for key import & establishment of trust
- **Volume Encryption**
  - Basic operation already present in RHEL5 – incremental centralized key management for RHEL 6
- **Sectool**
  - Compliance checking / intrusion detection utility – validates system admin config: file permissions, valid UIDs, reasonable passwords, etc.

s390-tools version 1.8.2

- This package provides *the* essential tool chain for Linux on System z. It contains everything from the boot loader to dump related tools for a system crash analysis.
- New Features
    - DASD related tools: Add Large Volume Support for ECKD DASDs
    - IPL tools: Can be used to change the reipl & shutdown behavior
    - ziomon tools: Set of tools to collect data for zfcp performance analysis.
    - lsluns: List available SCSI LUNs depending on adapter or port.
    - lszcrypt/chzcrypt: Show/modify information about zcrypt devices and configuration

s390-tools version 1.8.2

- New Features (continued)
  - cpuplugd: Daemon that manages CPU- and memory-resources based on a set of rules. Depending on the workload CPUs can be enabled or disabled. The amount of memory can be increased or decreased exploiting the Cooperative Memory Management (CMM1) feature.
  - lschp/chchp: Tool to show/modify channel-path states and available channel paths.
  - mon_procd: Daemon that writes process information data to the z/VM monitor stream.
  - vmur: Tool to work with z/VM spool file queues (reader, punch, printer).
  - zfcpdump_v2: Version 2 of the zfcpdump tool. Now based on the upstream Linux kernel 2.6.23.
- Plus various bug fixes

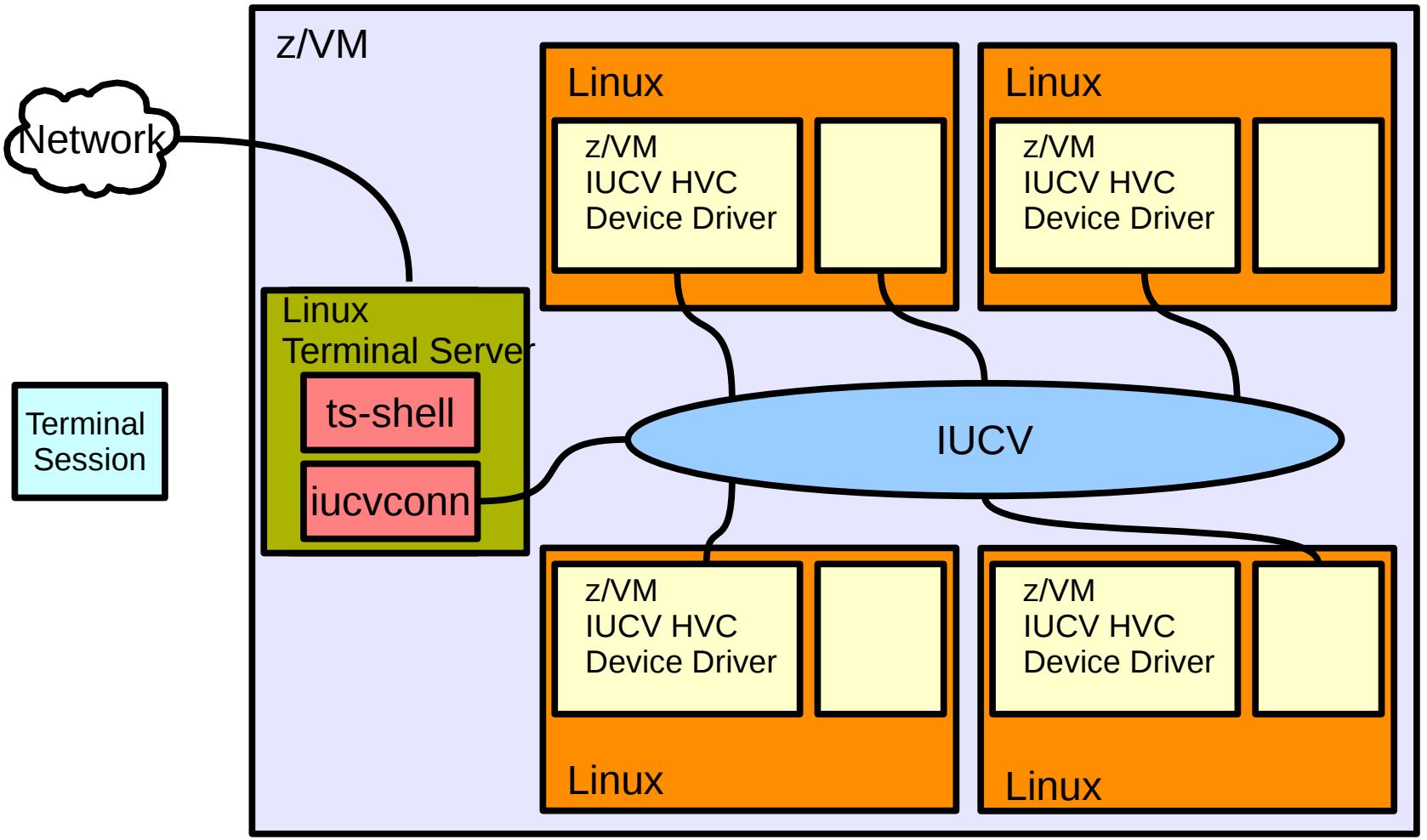
s390-tools version 1.8.2

- For more information, see Hans-Joachim Picht's Session
  - Session 8647
  - **s390-tools: The Swiss Army Knife for Linux on z System Administration**
  - Tuesday 3/1, 1:30 p.m. - 2:30 p.m, This room (203b)

Advanced Configurations: TTY Terminal

- **TTY Terminal Server Over IUCV**
  - Provide central access to the Linux console for the different guests of a z/VM.
  - The terminal server connects to the different guests over IUCV.
  - The IUCV based console is ASCII based.
  - Fullscreen applications like *vi* are usable on the console.

TTY Terminal Server Over IUCV

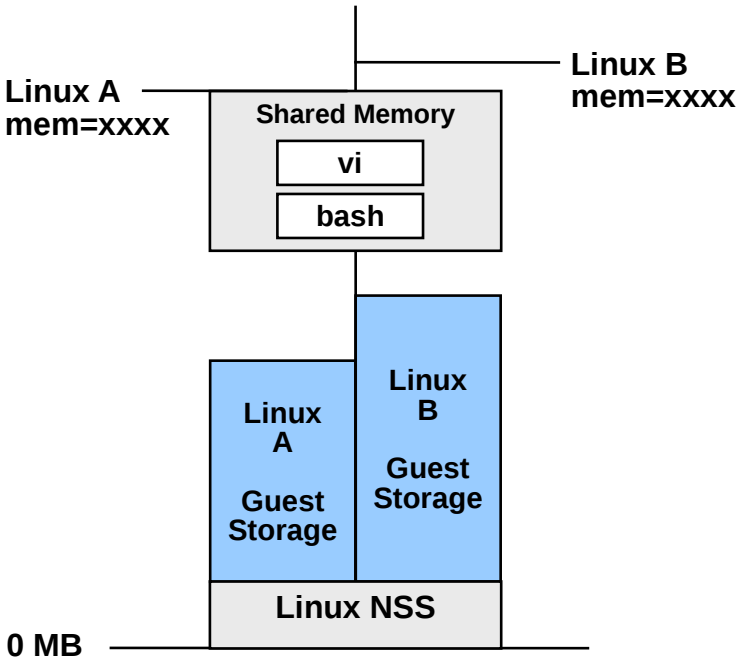


TTY Terminal Over IUCV

- For more information, see Hans-Joachim Picht's Session
  - Session 8463
  - **Introduction to the new Linux on System z Terminal Server using IUCV**
  - Today (Monday 2/28), 3:00 p.m. - 4:00 p.m, This room (203b)

Advanced Configurations: Shared Kernel

- **Named Saved Segments (NSS)**
  - Using NSS the z/VM hypervisior makes operating system code in shared real memory pages available to z/VM guest virtual machines.
  - With this update, Linux guest operating systems using z/VM can boot from the NSS and be run from a single copy of the Linux kernel in memory.



Benchmark Study: A Performance Comparison Between RHEL 5 and RHEL 6 on System z

Brad Hinson <bhinson@redhat.com>

Worldwide System z Sales, Strategy, Marketing

Lab Environment Setup

- Hardware and z/VM Environment
  - z10 EC, 4 IFLs used (non concurrent tests)
  - Separate LPARs for RHEL 5, RHEL 6
  - z/VM 6.1
  - IBM DS8000 Storage Array
    - ECKD DASD, mod 9
    - FCP through Fiber Switch
  - VSWITCH used for networking
- RHEL 5.5 GA, default install
- RHEL 6.0 GA, default install

Case 1: System Startup

- Test plan
  - Measure System startup using default options, with similar services configured
  - Leave as many defaults as possible

- Technical details
- /etc/rc.local: Last startup file to get executed after system boot
  - Add these lines:
    - echo "Startup complete:" >> /tmp/startup.test
    - date >> /tmp/startup.test
  - Now reboot with this command:
    - date > /tmp/startup.test; reboot

Case 1: System Startup

- Results
  - RHEL 5.5
    - **1 minute, 12 seconds**
  - RHEL 6.0
    - **34 seconds**
  - Over twice as fast (actual: 212%)

- Reasons
  - RHEL 6 has “upstart”, an event-based replacement for the /sbin/init daemon which handles starting of tasks and services during boot, stopping them during shutdown and supervising them while the system is running. RHEL 5 starts services in order (serially, one after another).
  - RHEL 6 default bootloader timeout has been reduced.

Case 2: Networking

- Test Plan
  - Use Apache web server benchmark to simulate heavy client network traffic
  - Client and server on same LPAR, using same VSWITCH, to avoid benchmarking external network speeds in lab

- Technical Details
  - 100,000 connections
  - 4 concurrent connections at a time
  - Payload size: small (170 bytes)
  - 3 test runs, then average results
  - RHEL 5.5: Apache version 2.2.3
  - RHEL 6.0: Apache version 2.2.15

Case 2: Networking

- Results

	RHEL 5.5	RHEL 6.0	% improvement
HTTP requests/sec	7289	9712	33%
Time per request	0.551 ms	0.413 ms	33%
Time per request, across all concurrent requests	0.138 ms	0.103 ms	34%
Transfer rate	3125 Kbytes/sec	4154 Kbytes/sec	33%

- Reasons
  - Network driver (qeth) redesigned. Also many kernel network subsystem improvements in RHEL 6.

Case 3: Sysbench

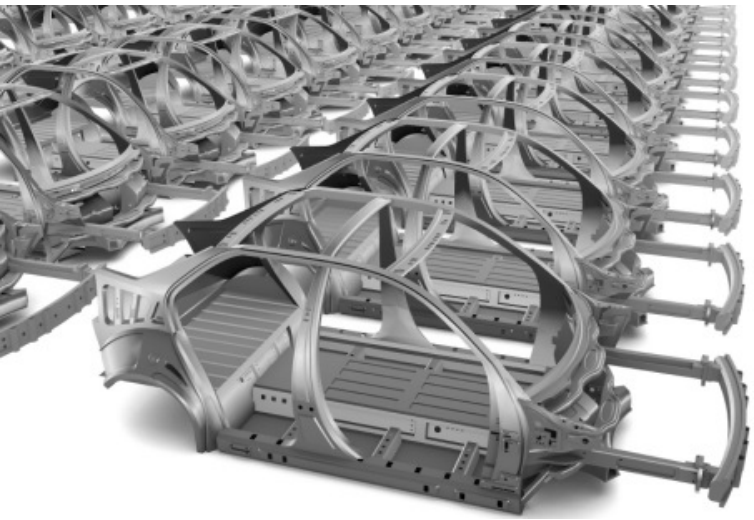
- Sysbench
  - *http://sysbench.sourceforge.net/*
  - “SysBench is a modular, cross-platform and multi-threaded benchmark tool for evaluating OS parameters that are important for a system running a database under intensive load. The idea of this benchmark suite is to quickly get an impression about system performance without setting up complex database benchmarks or even without installing a database at all.”

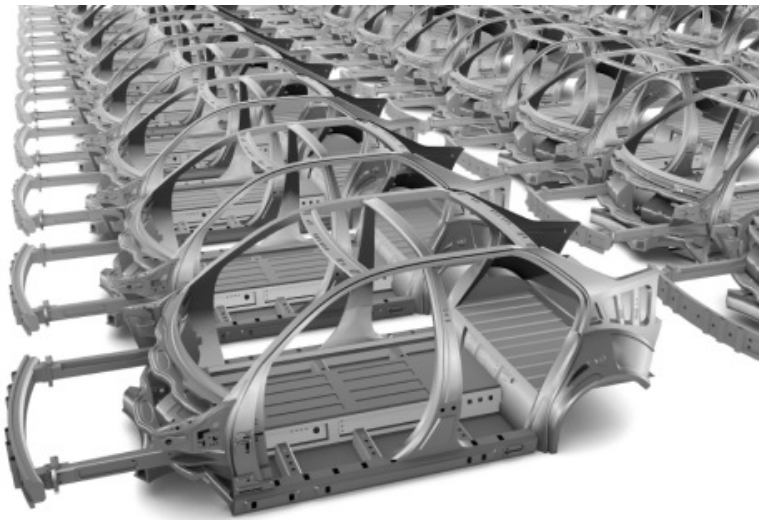
- Technical Details
  - Download and install sysbench package from Fedora
  - Rebuild source RPM required on RHEL 5

Case 3: Sysbench: Application Scheduler

- Test 1: Linux Kernel Scheduler
  - SysBench creates a specified number of threads and a specified number of locks. Then each thread starts running to compete for the locks. The more iterations are performed, the more concurrency is placed on the system.

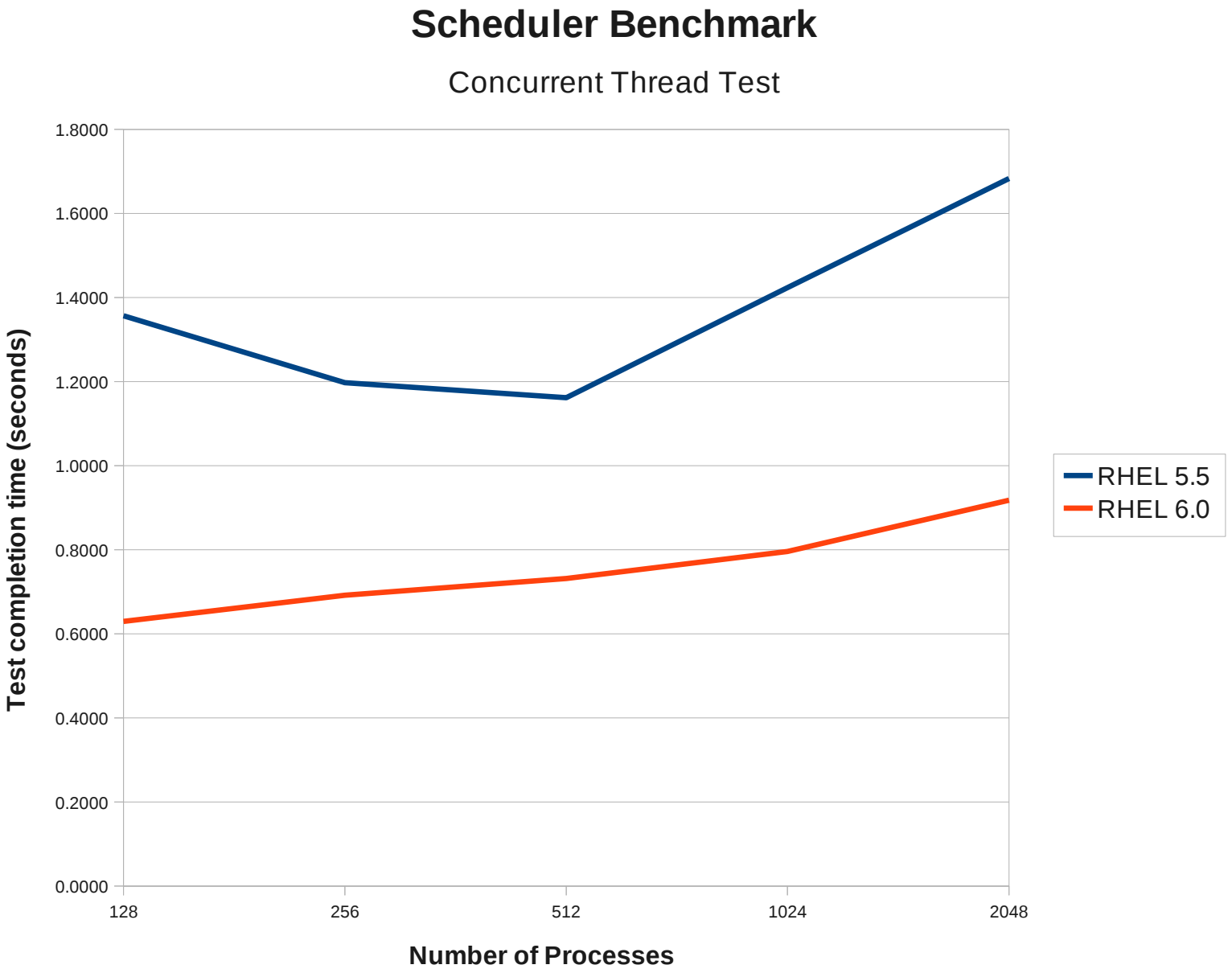
- Background:
  - RHEL 5 introduced the O(1) scheduler in 2007, which scaled better than kernel schedulers of the past.
  - Today, RHEL 6 uses the Completely Fair Scheduler (CFS), which is designed for multi-core CPU topologies, and operates with nanosecond granularity to maximize CPU utilization, and with lower power consumption.





Case 3: Sysbench: Application Scheduler

- Results:



Case 3: Sysbench: Memory Performance

- Test 2: Memory reads and writes
  - This is a simple test of reads and writes to memory.
  - Test memory size in all tests is 8GB. Guest memory size is 512MB.

- Results

	RHEL 5.5	RHEL 6.0	% improvement
Write Speed	1295 MB/s	2019 MB/s	56%
Read Speed	2471 MB/s	7735 MB/s	213%

- Reasons
  - Major improvements to the Linux kernel memory subsystem between RHEL 5 and RHEL 6.

Case 3: Sysbench: Disk I/O Performance

- Test 3: File I/O
  - This test simulates I/O workload to disk. The test was run on both ECKD Mod-9 DASD.

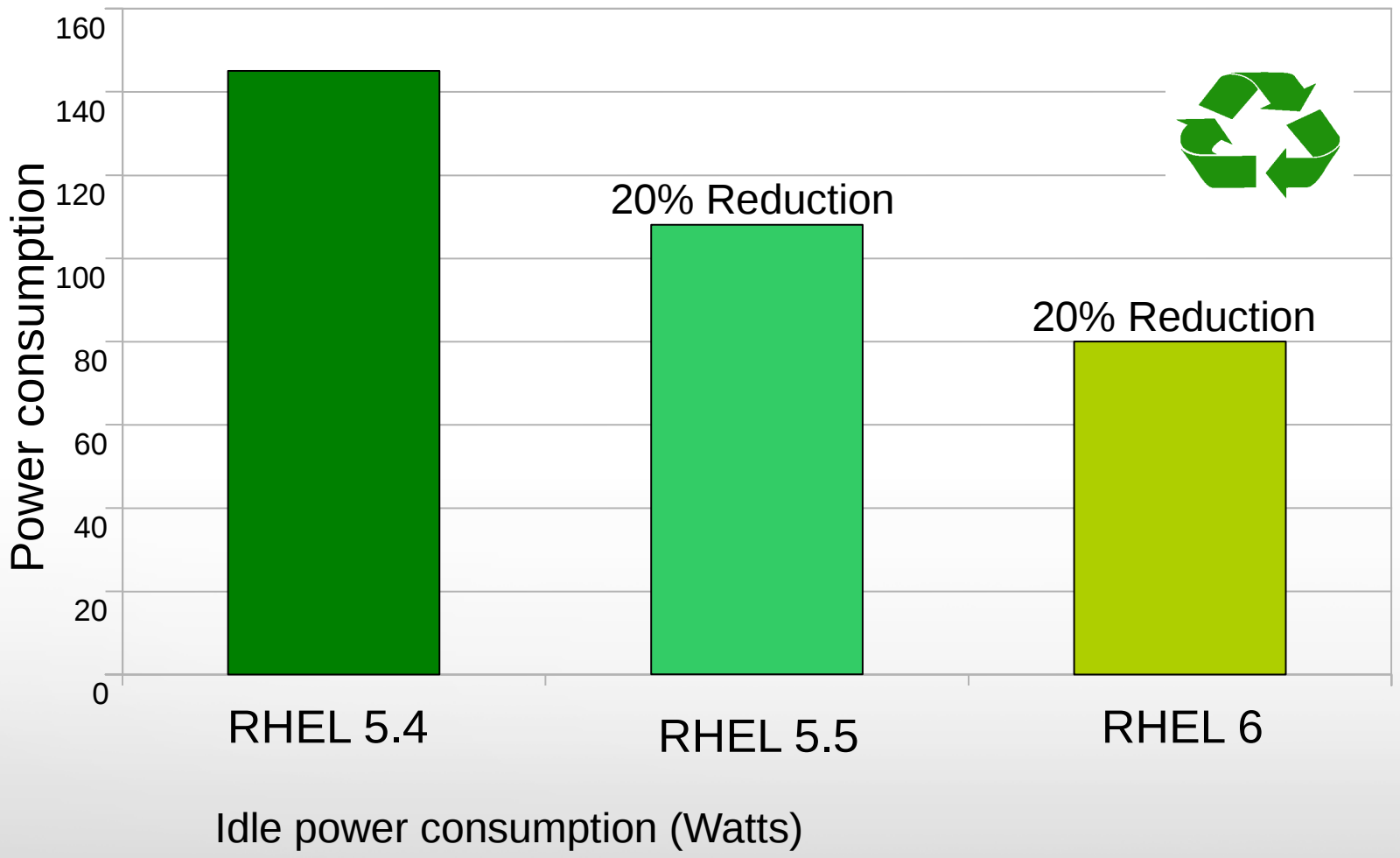
- Results

	RHEL 5.5 (ext3)	RHEL 6.0 (ext4)	% improvement
ECKD data operations/sec	963.29	1708.88 MB/s	77%
ECKD total speed	15.051 MB/s	26.701 MB/s	77%

- Reasons
  - Improvements to the DASD driver, as well as file system improvements in ext4, which is now the default in RHEL 6. Compared to ext3, the ext4 file system is faster, more robust, and supports extended functionality like thin provisioning.

Case 4: Run Leaner

Lower Power Consumption



Documentation Links

- Redbook, “z/VM and Linux on IBM System z: The Virtualization Cookbook for Red Hat Enterprise Linux 6”
  - http://www.redbooks.ibm.com/abstracts/sg247932.html
- DeveloperWorks**:**
  - http://www.ibm.com/developerworks/linux/linux390/documentation_dev.html
- Knowledgebase:
  - http://kbase.redhat.com/
  - Search “s390”
- http://www.redhat.com/z
- http://www.redhat.com/rhel/server/mainframe/promo/

Contact Info

- Brad Hinson, Red Hat

  Worldwide Lead, Linux on System z

  bhinson@redhat.com

  (919) 360-0443

redhat.

Questions?

Appendix:
RHEL 6 on System z, “Below the Line”

Advanced Virtualization

- **Dynamic Memory Add/Remove** (kernel 2.6.27)
  - Attach and use standby memory that is configured for a logical partition or z/VM guest.
  - Memory Attach & Detach requires running Linux on System z as a VM-guest requires z/VM 5.4 plus the PTF for APAR VM64524.
- **Standby CPU Activation/Deactivation** (kernel 2.6.25)
  - Allow standby CPUs to be activated / deactivated
- **Extra Kernel Parameter for SCSI IPL** (kernel 2.6.32)
  - Modify the SCSI loader to append append extra parameters specified with the z/VM VMPARM option to the kernel command line.

redhat

Advanced Virtualization

- **hvc_iucv: Provide IUCV z/VM User ID Filtering** (kernel 2.6.29)
  - Introduces the kernel parameter "hvc_iucv_allow=" that specifies a comma-separated list of z/VM user IDs.
  - If specified, the z/VM IUCV hypervisor console device driver accepts IUCV connections from listed z/VM user IDs only.
- **Suspend / Resume** (kernel 2.6.31)
  - With suspend and resume support, you can stop a running Linux on System z instance and later continue operations.
  - When Linux is suspended, data is written to a swap partition. The resume process uses this data to make Linux continue from where it left off when it was suspended.
  - A suspended Linux instance does not require memory or processor cycles.

redhat

Suspend/Resume Support

- Ability to stop a running Linux on System z instance and later continue operations
- Memory image is stored on the swap device specified with a kernel parameter: `resume=/dev/dasd<x>`
- ```
  # grep swap /etc/fstab
  /dev/dasdd1 swap swap pri=-1 0 0
  /dev/dasde1 swap swap pri=-2 0 0
  ```
- Suspend operation is started with “echo” command to sysfs:

```
# echo disk > /sys/power/state
```

- Resume is done automatically on IPL
- Use signal (Ctrl+Alt+Del) to automatically suspend a guest:

```
ca::ctrlaltdel:/bin/sh -c "/bin/echo disk > \
/sys/power/state || /sbin/shutdown -t3 -h now"
```
```

Storage Support

- **FCP Automated Port Discovery** (kernel 2.6.25)
  - Scan the connected fiber channel SAN and automatically activate all available and accessible target ports. This requires a proper SAN setup with zoning.
- **FCP SCSI Error Recovery Hardening** (kernel 2.6.30)
  - Avoid SCSI error recovery escalation in case of concurrent zfcp and SCSI error recovery.
- **FCP Adjustable Queue Depth** (kernel 2.6.31)
  - Customizable queue depth for SCSI commands in zfcp.

Storage Support

- **HyperPav** (kernel 2.6.25)
  - HyperPav is addressing the need to access more data with good performance and high availability!
  - This feature, which required a IBM DS8000™ disk storage system in average leads to a higher utilization, resulting in I/O transfer rates.
  - Activated automatically when the necessary prerequisites are there (DS8000 with HyperPAV LIC, z/VM 5.3). Transparent for the Linux on System z guest
- **I/O Configuration Support** (kernel 2.6.27)
  - Adds the infrastructure to allow Linux system to change the I/O configuration of a System z system.
  - Operations are addition, removal and reconfiguration/reassignment of I/O channels, control units and subchannels.
  - This support is available only when running Linux on System z in an LPAR.

Storage Support

- **High Performance FICON (HPF)** (kernel 2.6.29)
  - Added HPF support to the DASD Device Driver
  - HPF is an extension to the FICON architecture and is designed to improve the execution of small block I/O requests.
  - HPF streamlines the FICON architecture and reduces the overhead on the channel processors, control unit ports, switch ports, and links by improving the way channel programs are written and processed.
- **FCP SCSI Error Recovery Hardening** (kernel 2.6.30)
  - Avoid SCSI error recovery escalation in case of concurrent zfcp and SCSI error recovery.
- **DASD Large Volume Support** (kernel 2.6.29)
  - Large Volume Support is a feature that allows to use ECKD devices with more than 65520 cylinders. This features is available with DS8000 R4.0

Networking

- **HiperSockets Network Traffic Analyzer** (kernel 2.6.33)
  - Trace HiperSockets network traffic for problem isolation and resolution. Supported for layer 2 and layer 3
- **Crypto Express 3** (kernel 2.6.33)
  - Support for Crypto Express3 Accelerator (CEX3A) and Crypto Express3 Coprocessor (CEX3C)
  - z/VM 6.1, 5.4, or 5.3 with the PTF for APAR VM64656 is required for z/VM guest-support.
- **Secondary Unicast Addresses** (kernel 2.6.27)
  - Allow secondary unicast MAC addresses to support MAC address based VLANs
  - This only works with an OSA interface running in layer 2 mode.

Networking

- **OSA QDIO Data Connection Isolation** (kernel 2.6.33)
  - Feature (available for OSA-Express2 cards since early 2009) allows an operating system to configure the OSA adapter to prevent any direct package exchange between itself and other operating system instances that share the same OSA adapter.
  - The configuration of the isolation level is done through sysfs.
  - Starting with s390-tools 1.8.4 lsqeth indicates connection isolation in qeth attribute 'isolation'.
- **QETH Componentization** (kernel 2.6.25)
  - The qeth driver module is split into a core module and layer2-/layer3-specific modules. The default operation mode for OSA-devices is changed to layer2; for HiperSockets devices the layer3 default-mode is kept.
  - For layer3 mode devices the existence of (possibly faked) ethernet-headers is guaranteed to enable smooth integration of qeth devices into Linux.

RAS: Reliability, Availability, Serviceability

- **Automatic IPL After Dump** (kernel 2.6.30)
  - Extension to the shutdown action interface which combines the actions dump and re-ipl, first a dump is taken, then a re-ipl of the system is triggered

- **Compiler Improvements** (gcc 4.4)
  - The latest compiler enhancements allow a customer to recompile existing applications which can be optimized for the latest hardware generation without any changes to the source code.
  - This can lead up to a > 10 % performance improvement.

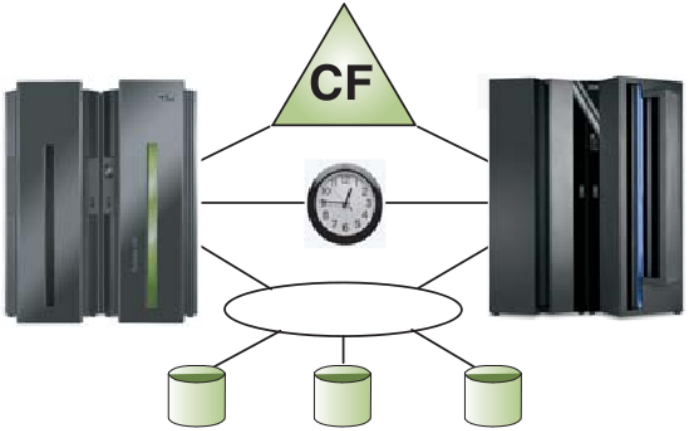
- **Large Page Support (**kernel 2.6.25)
  - Support for a new access method to allocate larger chunks of memory (1 MB page size), resulting in performance improvements, especially in Java based environments
  - This feature exploits z10 hardware features and provides a software emulation for older systems.

RAS: Reliability, Availability, Serviceability

- **Add Call Home Data on Halt and Panic if Running in LPAR** (kernel 2.6.32)
  - Report system failures (kernel panic) via the service element to the IBM service organization.
  - Improves service for customers with a corresponding service contract. (by default this features is deactivated)
- **CIO, DASD: Improved DASD Error Recovery** (2.6.33)
  - Improved the DASD error recovery procedures used in the early phases of IPL and DASD device initialization.

Miscellaneous

- **Kernel VDSO Support** (kernel 2.6.29)
  - Kernel provided shared library to speed up a few system calls (gettimeofday, clock_getres, clock_gettime)
- **Kernel Image Compression** (kernel 2.6.33)
  - The kernel image size can be reduced by using one of three compression algorithms: gzip, bzip2 or lzma.
- **STP/ETR Support** (kernel 2.6.27)
  - Support for clock synchronization using the server time protocol (STP) or an external



Miscellaneous

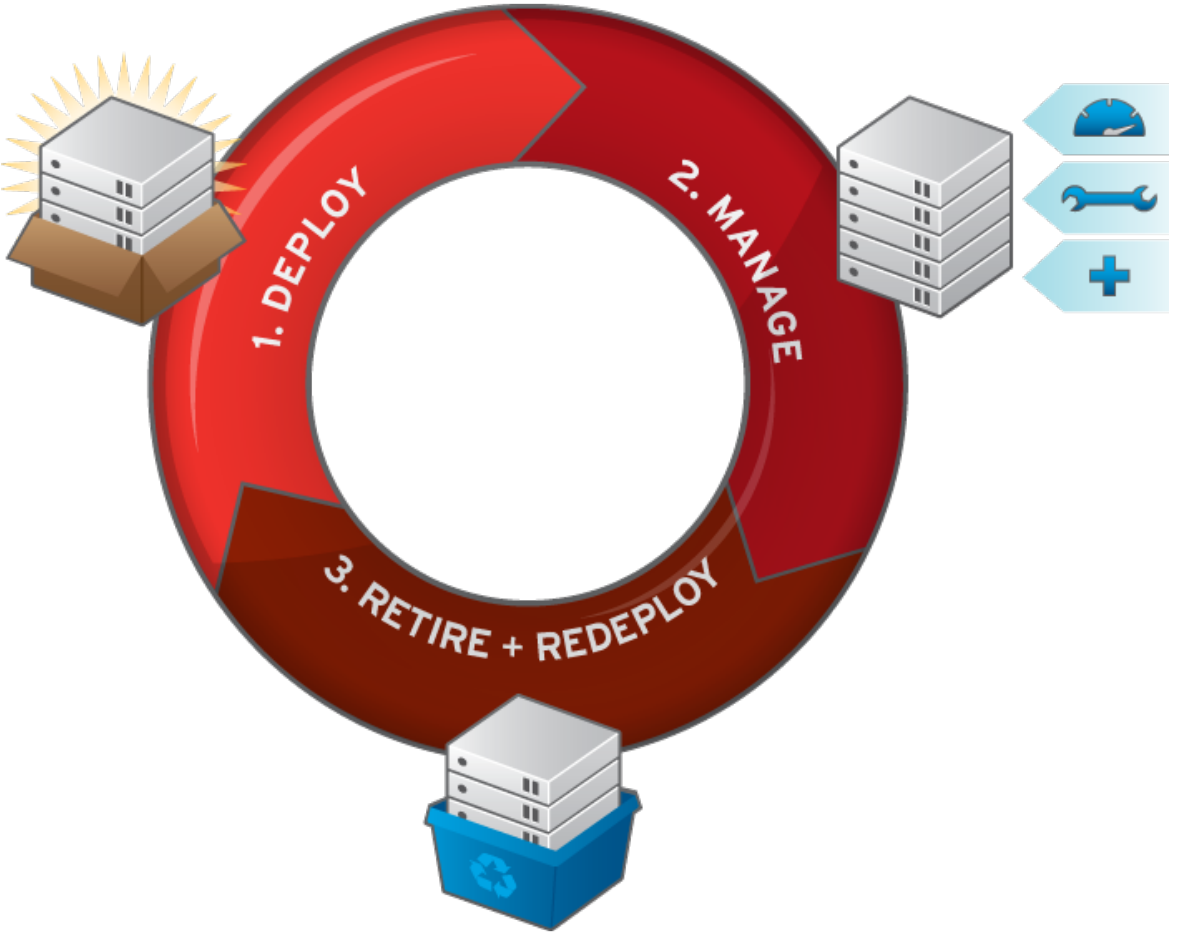
- **KULI (2009-06-24)**
  - kuli is experimental userspace sample to demonstrate that KVM can be used to run virtual machines on Linux on System z.
  - This experimental proof of concept is unsupported and should not be used for any production purposes.
- **Oprofile**
  - Starting with version 0.9.4, oprofile supports sampling of Java byte code applications for Linux on System z.
- **Eclipse 3.3**
  - Starting with Eclipse 3.3, Linux on System z is officially supported.

Kernel Scalibility

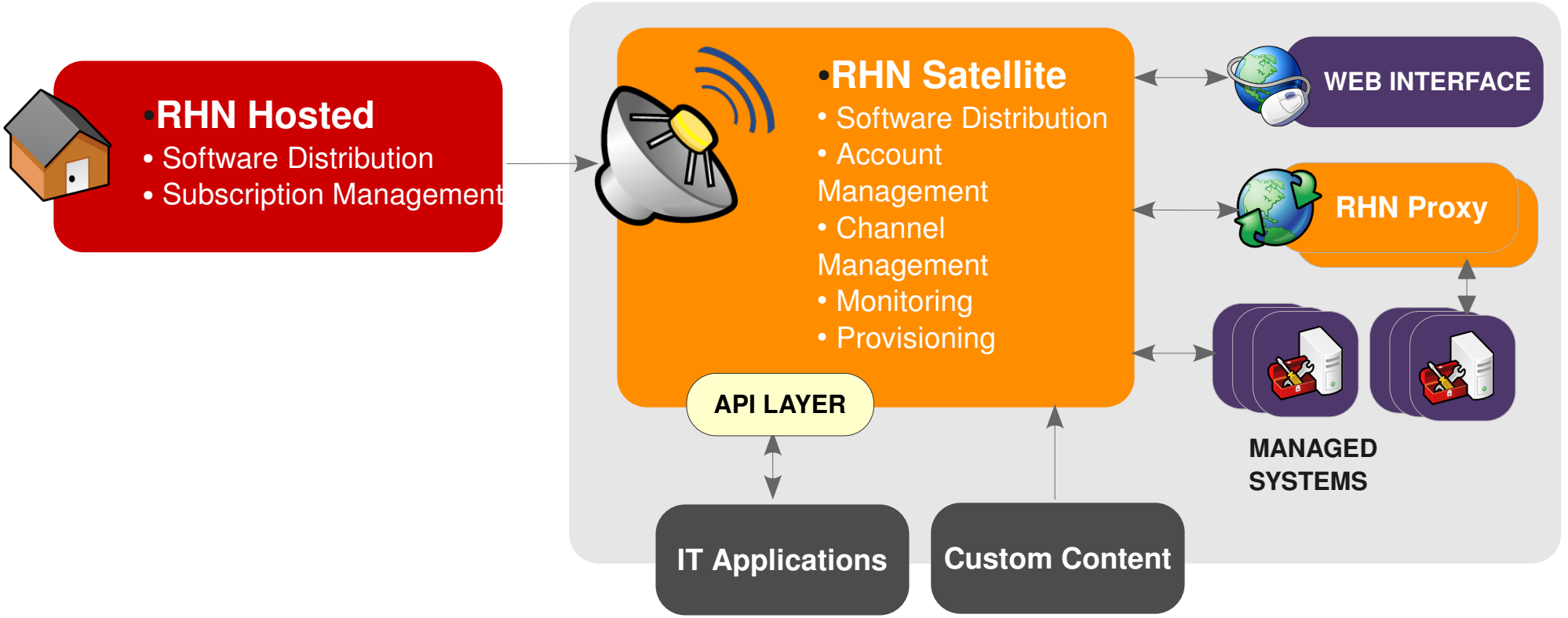
- **Scalability Limits – Maximum Values**
  - Max CPUs – dynamic allocation of CPU structs (2.6.29) allows supported limit of 4096 and theoretical limit of 64K.
  - Max IRQs – dynamic allocation (2.6.28) allows limit of 256
  - Max memory 48-bit addressing (256TB) requires pending patch incorporation
  - Max # processes – 4 million on 64 bit kernels
  - Max threads per process remains at 32000 (same as RHEL5)
  - Filesystem limits for ext4 – 100T (practical target – bounded by fsck time)

- **Scalability Features**
  - Split LRU VM – different eviction policies for file backed vs swap backed

Appendix:
Systems Management
with RHN Satellite

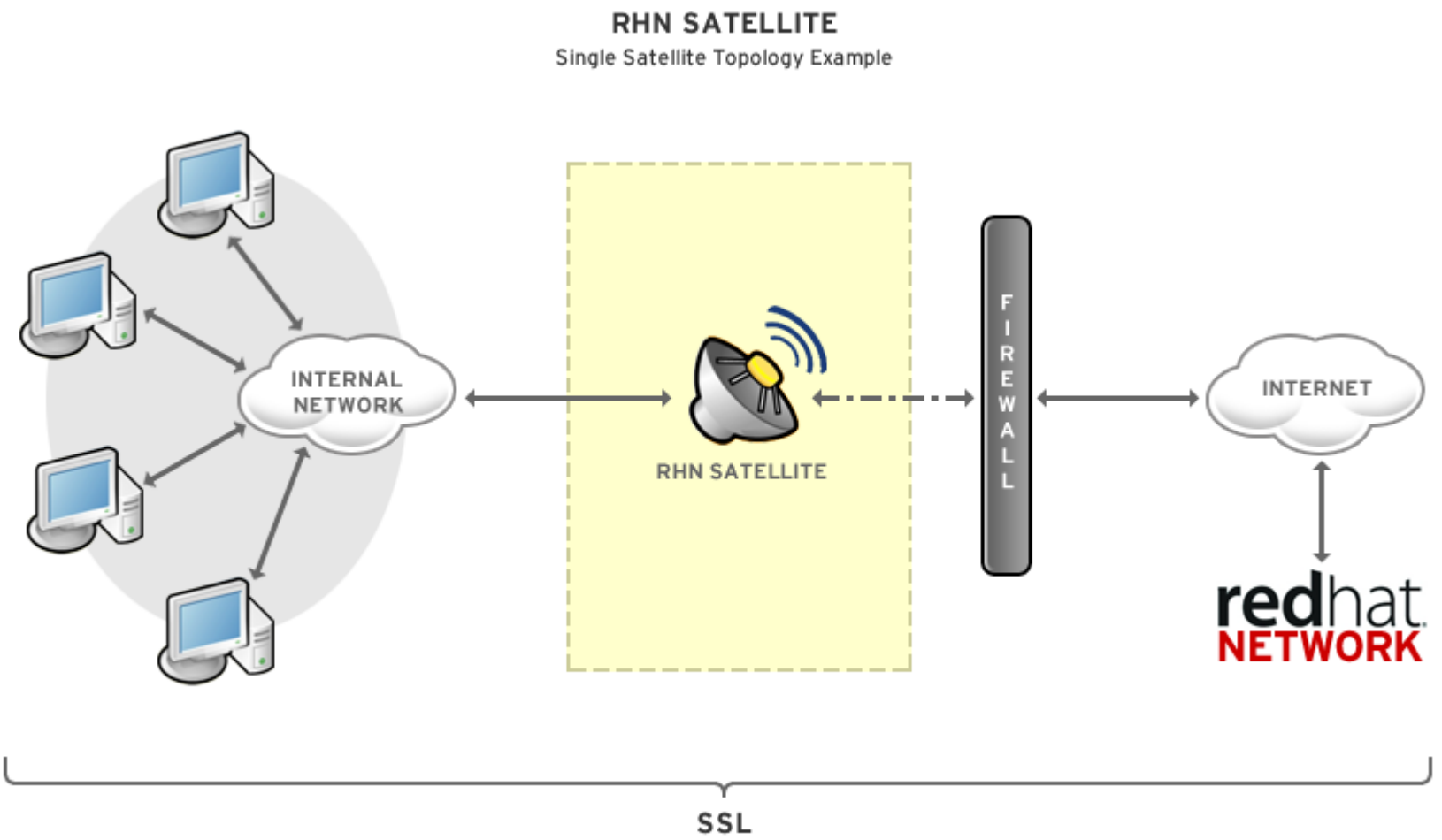


Satellite deployment model



- Enterprise management solution – enhanced control
- Local database stores all packages, profiles, and system information
- Syncs content from RHN Hosted
- Custom content distribution
- Can run disconnected from the Internet

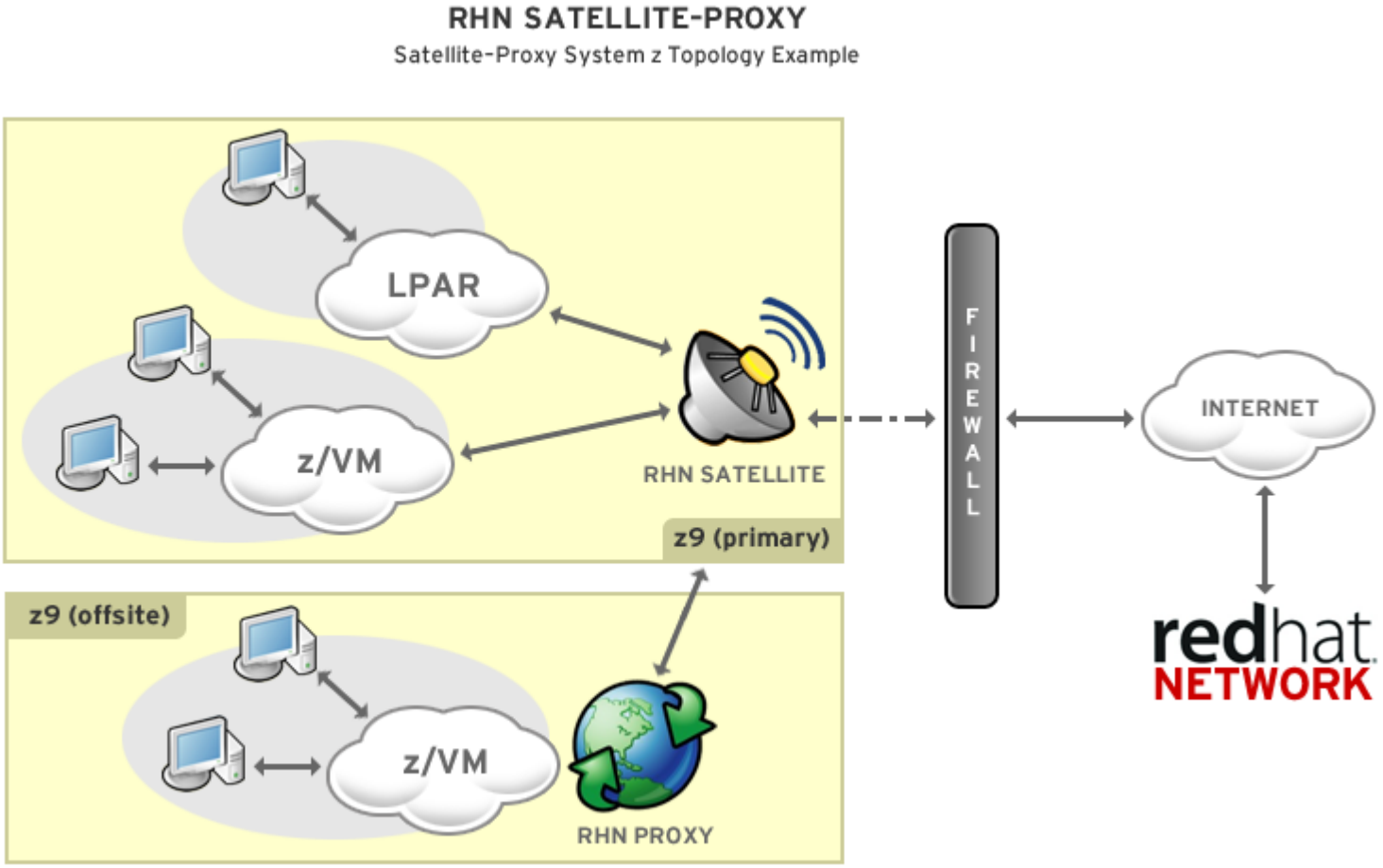
RHN Satellite



redhat

RED HAT NETWORK
:: Enterprise Systems Management

RHN Satellite on System z



PXE Deployment on System z

zPXE

- Same configuration/profile on all guests
- Read-write 191 disk not required for each guest
- All changes kept on management server
- Flexibility of kickstart
- Same principles as traditional PXE, adapted to System z
- Fits with configuration management tools (cobbler)
- Easy to update

https://fedorahosted.org/cobbler/wiki/SssThreeNinety

redhat.

Thank You