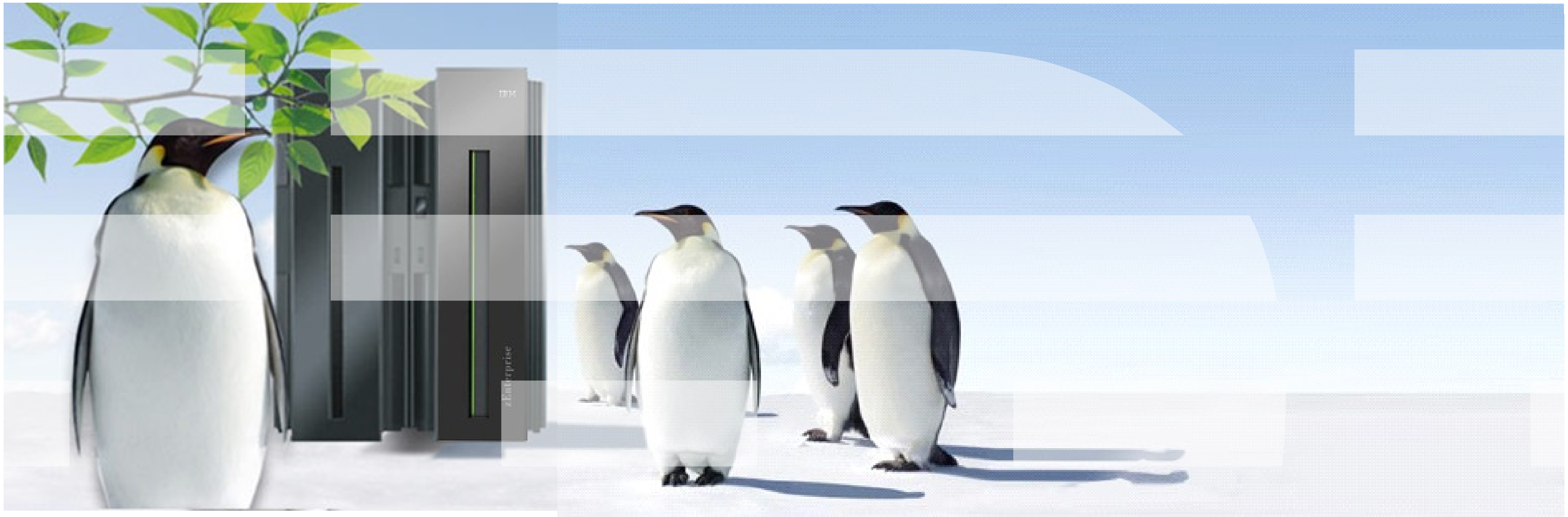
What's New in Linux on System z

Martin Schwidefsky
IBM Lab Böblingen, Germany

February 6 2013



Trademarks & Disclaimer

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

IBM, the IBM logo, BladeCenter, Calibrated Vectored Cooling, ClusterProven, Cool Blue, POWER, PowerExecutive, Predictive Failure Analysis, ServerProven, System p, System Storage, System x , System z, WebSphere, DB2 and Tivoli are trademarks of IBM Corporation in the United States and/or other countries. For a list of additional IBM trademarks, please see http://ibm.com/legal/copytrade.shtml.

The following are trademarks or registered trademarks of other companies: Java and all Java based trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries or both Microsoft, Windows,Windows NT and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both. Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries or both. Linux is a trademark of Linus Torvalds in the United States, other countries, or both. Cell Broadband Engine is a trademark of Sony Computer Entertainment Inc. InfiniBand is a trademark of the InfiniBand Trade Association.
Other company, product, or service names may be trademarks or service marks of others.

NOTES: Linux penguin image courtesy of Larry Ewing (lewing@isc.tamu.edu) and The GIMP

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Users of this document should verify the applicable data for their specific environment. IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
Information is provided “AS IS” without warranty of any kind. All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices are suggested US list prices and are subject to change without notice. Starting price may not include a hard drive, operating system or other features.
Contact your IBM representative or Business Partner for the most current pricing in your geography. Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use. The information could include technical inaccuracies or typographical errors.
Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any

How Linux on System z is developed

How does the “community” work.

IBM collaborates with the Linux community

- has been an active participant since 1999
- is one of the leading commercial contributors to Linux
- has over 600 full-time developers working with Linux and open source

Linux Kernel & Subsystem Development

Kernel Base

Security

Systems Mgmt

Virtualization

Filesystems,

and more...

Expanding the Open Source Ecosystem

Apache

Eclipse

Mozilla Firefox

OpenOffice.org,

and more...

Promoting Open Standards & Community Collaboration

The Linux Foundation

Linux Standards Base

Common Criteria certification,

and more...

Foster and Protect the Ecosystem

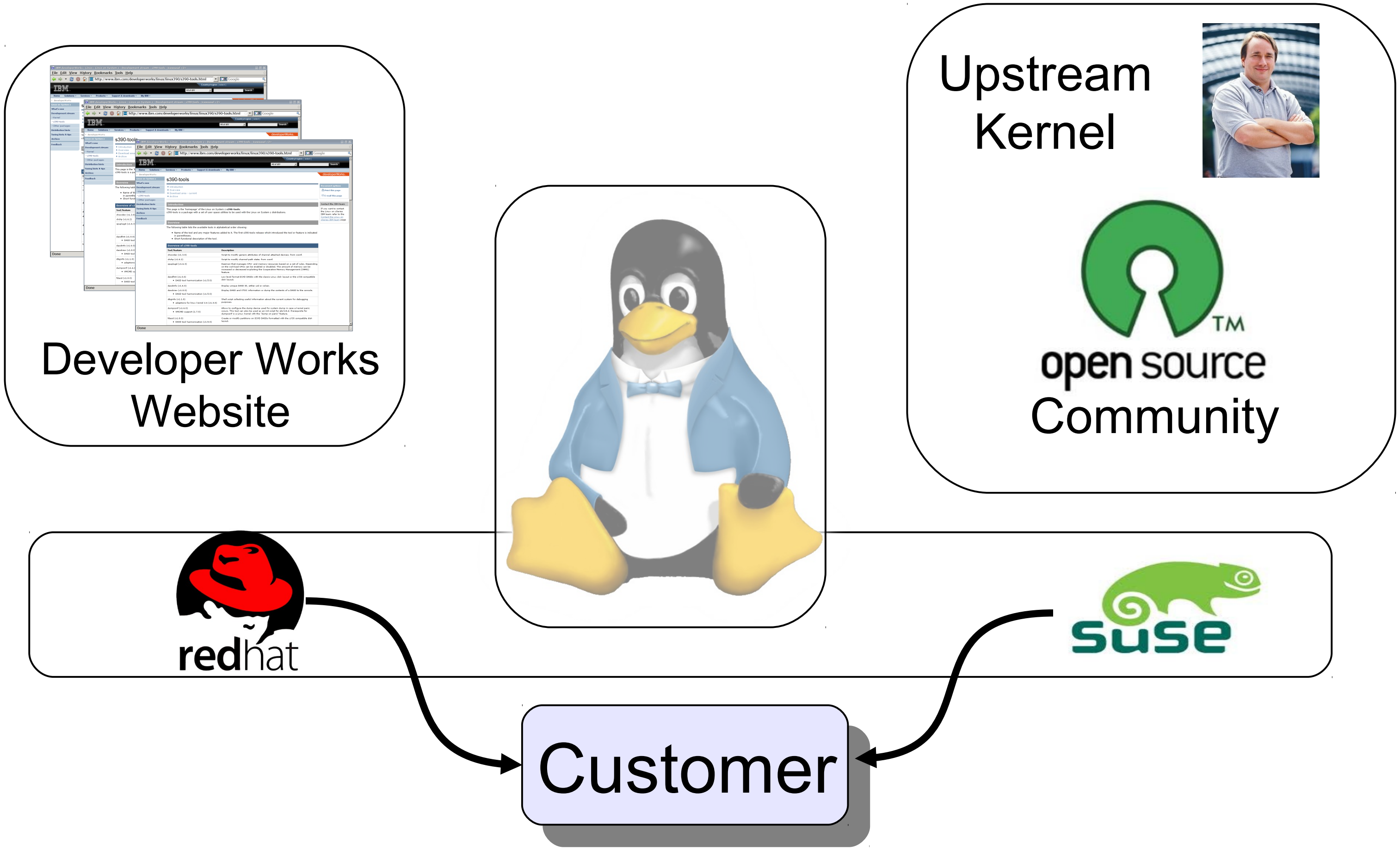
Software Freedom Law Center

Free Software Foundation (FSF),

and more...

The IBM Linux development process

IBM Linux on System z development contribues in the following areas: Kernel, s390-tools, open source tools (e.g. eclipse, ooprofile), gcc, glibc, binutils



Developer Works Website

Upstream Kernel

Community

Customer

Facts on Linux

- Linux kernel 1.0.0 was released with 176,250 lines of code
  How many lines of code has the kernel version 3.7 ?

  16191784 lines of code (+1,193,047 since v3.2)

- How many of the world's top 500 supercomputers run Linux (Nov 2012)?

  453 / 90.6%

- What percentage of web servers run Linux (Nov 2012) ?

  64.9% run Unix, of those 50.7% run Linux (47.5% unknown) = 32.9%

- What percentage of desktop clients run Linux (Dec 2012) ?

  1.75%

- What is the largest Linux architecture in number of devices ?

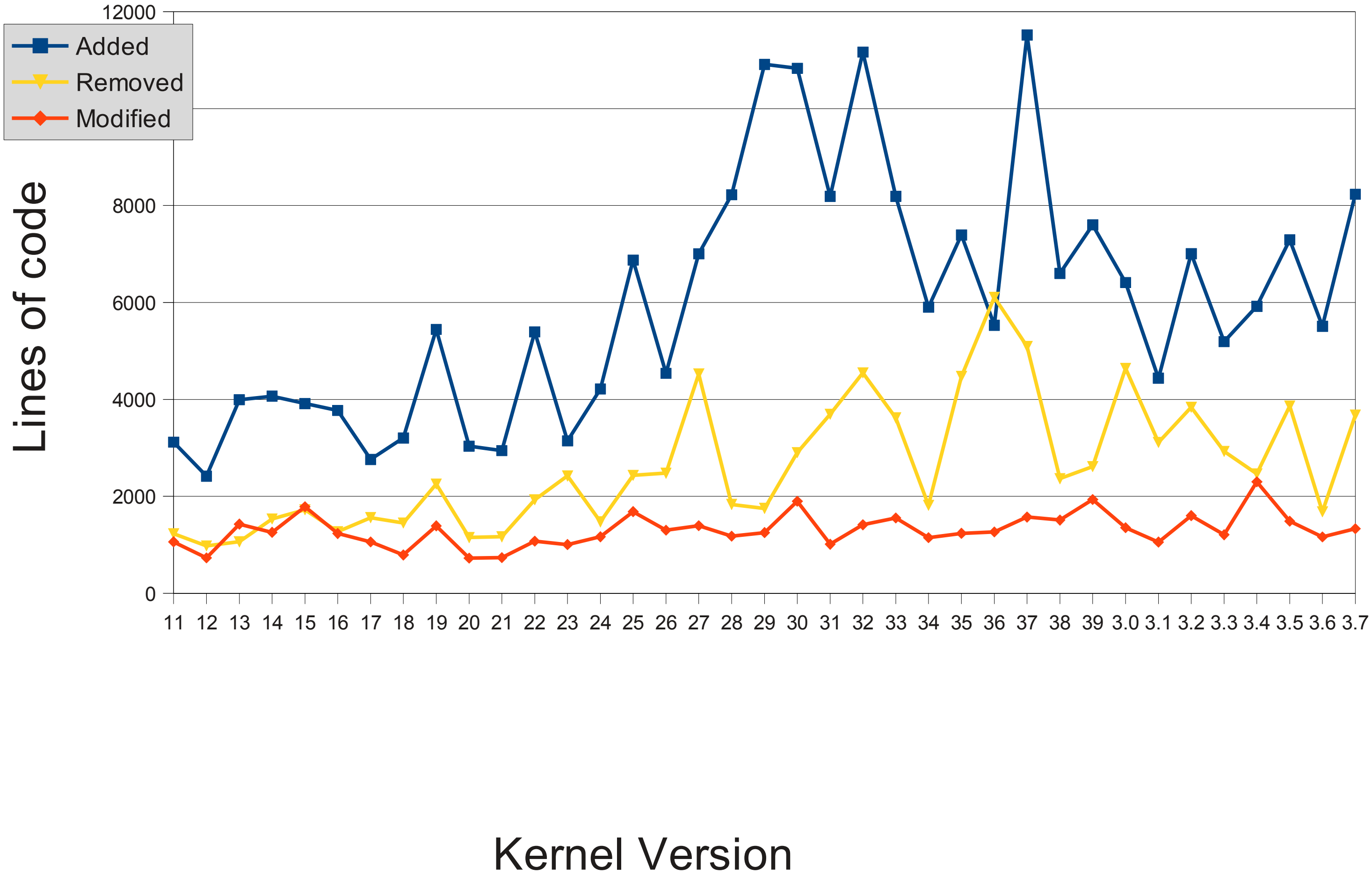
  ARM, > 100 million activated android devices

- Linux is Linux, but …features, properties and quality differ dependent on your platform and your use case

Source: http://kernel.org
http://top500.org/statistics
http://w3techs.com
http://www.w3counter.com
http://googleblog.blogspot.com/2011/05/android-momentum-mobile-and-more-at.html

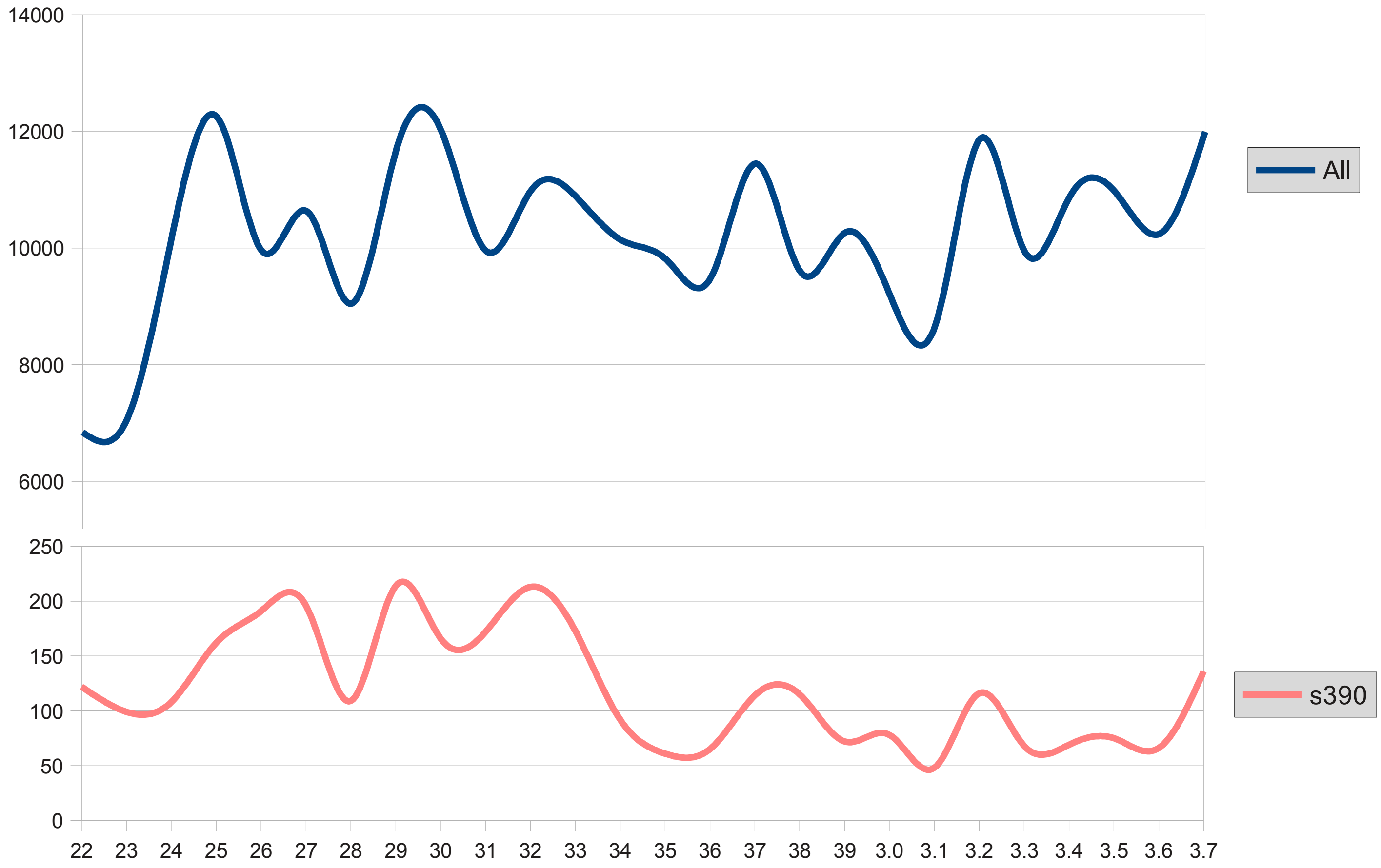
Linux kernel development: rate of change

Average for the last 7 3/4 years (without renames): 78 days per release, 5955 lines added, 2623 lines removed and 1251 lines modified **per day**



Linux kernel development: System z contributions

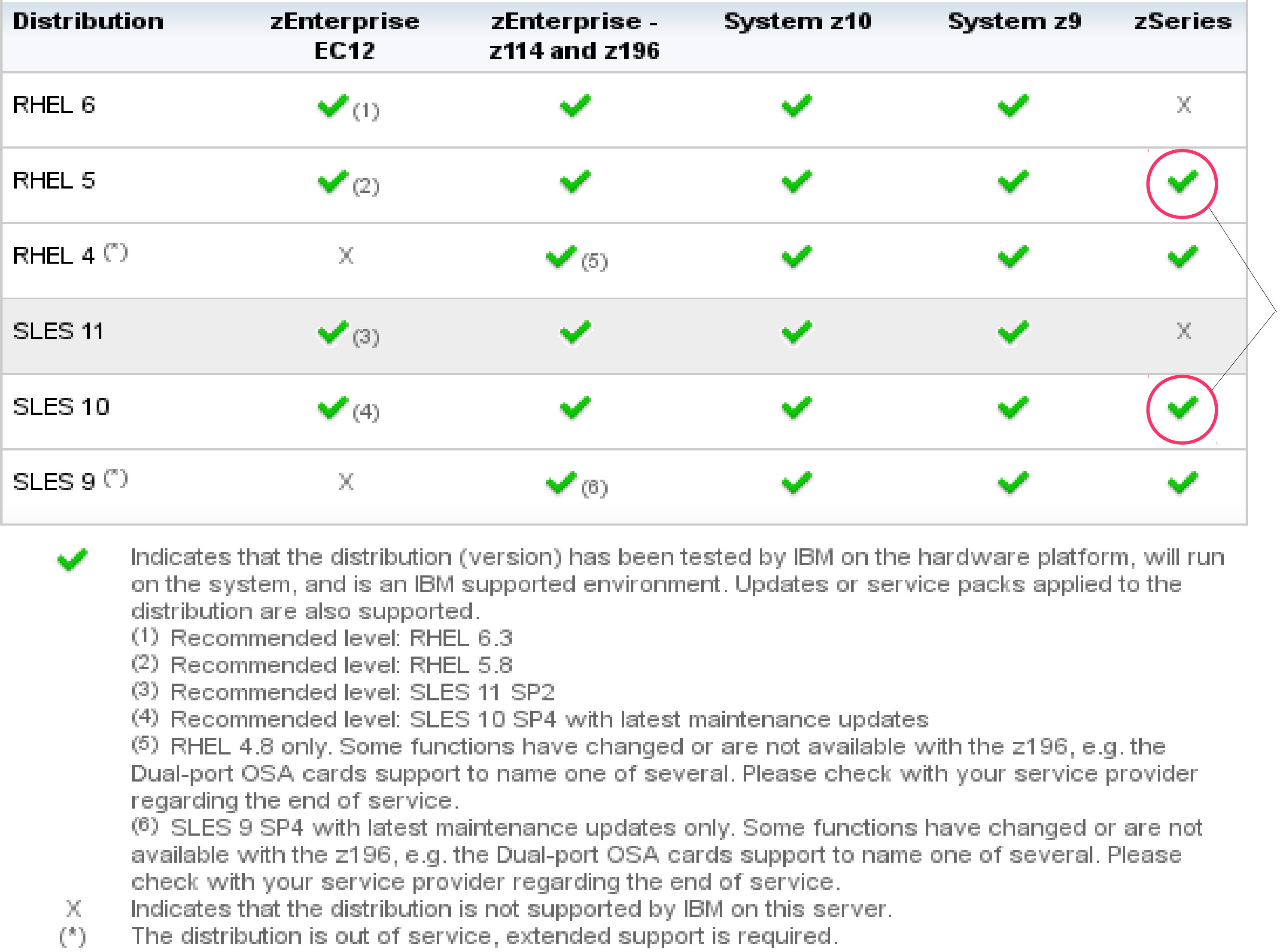
Changesets per 2.6.x / 3.x kernel release



Linux on System z distributions (Kernel 2.6 based)

- **SUSE Linux Enterprise Server 9 (GA 08/2004)**
  - Kernel 2.6.5, GCC 3.3.3, Service Pack 4 (GA 12/2007), end of regular life cycle
- **SUSE Linux Enterprise Server 10 (GA 07/2006)**
  - Kernel 2.6.16, GCC 4.1.0, Service Pack 4 (GA 05/2011)
- **SUSE Linux Enterprise Server 11 (GA 03/2009)**
  - Kernel 2.6.27, GCC 4.3.3, Service Pack 1 (GA 06/2010), Kernel 2.6.32
  - Kernel 3.0.13, GCC 4.3.4, Service Pack 2 (GA 02/2012)
- **Red Hat Enterprise Linux AS 4 (GA 02/2005)**
  - Kernel 2.6.9, GCC 3.4.3, Update 9 (GA 02/2011), end of regular life cycle
- **Red Hat Enterprise Linux AS 5 (GA 03/2007)**
  - Kernel 2.6.18, GCC 4.1.0, Update 9 (GA 01/2013)
- **Red Hat Enterprise Linux AS 6 (GA 11/2010)**
  - Kernel 2.6.32, GCC 4.4.0 Update 3 (GA 05/2012)
- **Others**
  - Debian, Slackware,
  - Support may be available by some third party

Supported Linux Distributions



Distribution	zEnterprise EC12	zEnterprise - z114 and z196	System z10	System z9	zSeries
RHEL 6	✓ (1)	✓	✓	✓	X
RHEL 5	✓ (2)	✓	✓	✓	✓
RHEL 4 (*)	X	✓ (5)	✓	✓	✓
SLES 11	✓ (3)	✓	✓	✓	X
SLES 10	✓ (4)	✓	✓	✓	✓
SLES 9 (*)	X	✓ (6)	✓	✓	✓

✓	Indicates that the distribution (version) has been tested by IBM on the hardware platform, will run on the system, and is an IBM supported environment. Updates or service packs applied to the distribution are also supported. (1) Recommended level: RHEL 6.3 (2) Recommended level: RHEL 5.8 (3) Recommended level: SLES 11 SP2 (4) Recommended level: SLES 10 SP4 with latest maintenance updates (5) RHEL 4.8 only. Some functions have changed or are not available with the z196, e.g. the Dual-port OSA cards support to name one of several. Please check with your service provider regarding the end of service. (6) SLES 9 SP4 with latest maintenance updates only. Some functions have changed or are not available with the z196, e.g. the Dual-port OSA cards support to name one of several. Please check with your service provider regarding the end of service.
X	Indicates that the distribution is not supported by IBM on this server.
(*)	The distribution is out of service, extended support is required.

Two options for zSeries machines

Current Linux on System z Technology

Features & Functionality developed in the past 2 years contained in the SUSE & Red Hat Distributions

Linux on System z features – Core kernel

- **z196 enhanced node affinity support (kernel 2.6.37)** 6.1 11.2
  - Allows the Linux scheduler to optimize its decisions based on the z196 topology
- **Performance indicator bytes (kernel 2.6.37)** 6.1 11.2
  - Display capacity adjustment indicator introduced with z196 via /proc/sysinfo
- **QDIO outbound scan algorithm (kernel 2.6.38)** 11.2
  - Improve scheduling of QDIO tasklets, OSA / HiperSockets / zfcp need different thresholds
- **Enabling spinning mutex (kernel 2.6.38)** 6.3 11.2
  - Make use of the common code for adaptive mutexes.
  - Add a new architecture primitive arch_mutex_cpu_relax to exploit sigp sense running to avoid the mutex lock retries if the hypervisor has not scheduled the cpu holding the mutex.
- **Detailed IRQ statistics (kernel 2.6.38)** 11.2
  - A detailed, per-cpu list of interrupts in /proc/interrupts
  - Useful to debug interrupt imbalances

Linux on System z features – Core kernel

11.2

- **Jump label support (kernel 3.0)**
  - Branch optimization for conditions that are rarely toggled, e.g. tracepoints

6.3

- **Two stage dumper / kdump support (kernel 3.2, s390-tools-1.17.0)**
  - Use a Linux kernel to create a system dump
    - Use a preloaded crashkernel to run in case of a system failure
    - Can be triggered either as panic action or by the stand-alone dumper, integrated into the shutdown actions framework
  - Pro
    - Enhanced dump support that is able to reduce dump size, shared disk space, dump to network, dump to a file-system etc.
    - The makedumpfile tool can be used to filter the memory of the crashed system
    - Dump disk space sharing is possible for server farms using network dump
  - Con
    - kdump is not as reliable as the stand-alone dump tools
    - kdump cannot dump a z/VM named saved system (NSS)
    - For systems running in LPAR kdump consumes memory

Liux on System z features – two stage dumper / kdump support

6.3

- Add a crashkernel= to the kernel command line

```
crashkernel=<size>@<offset>
```

- Boot your system and check the reservation

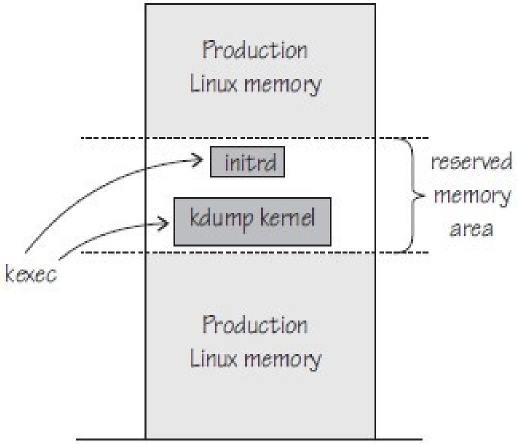
```
#cat /proc/iomem
00000000-3fffffff : System RAM
00000000-005f1143 : Kernel code
005f1144-00966497 : Kernel data
00b66000-014c4e9f : Kernel bss
40000000-47ffffff : Crash kernel
48000000-7fffffff : System RAM
```

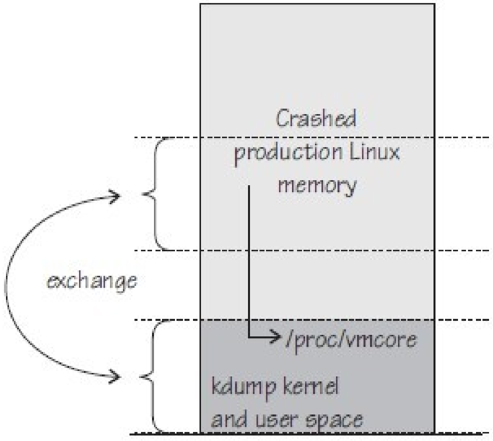
- Load the kdump kernel with kexec

```
# kexec -p kdump.image –initrd kdump.initrd
--command-line=”dasd=1234 root=/dev/ram0”
```

- Manually trigger for kdump under z/VM

```
#cp system restart
```





Linux on System z features – Usability / RAS

- **hyptop (kernel 2.6.35, s390-tools 1.12.0)** 11.2
  - A top-like tool that displays a dynamic real-time view of the hypervisor environment
  - It works with both the z/VM and the LPAR hypervisor
- **Address space randomization (kernel 2.6.38)** 6.2 11.2
  - Enable flexible mmap layout for 64 bit to randomize start address for the runtime stack and the mmap area
- **Get CPC name (kernel 2.6.39)** 6.2 11.2
  - Useful to identify a particular hardware system in a cluster
  - The CPC name and the HMC network name are provided
- **Module read-only protection (kernel 2.6.39)** 11.2
  - The text and the read-only data section of modules are write protected in the kernel page table, this avoids unwanted modification

Linux on System z features – FICON

- **Query DASD reservation status (kernel 2.6.37)** 6.1 11.2
  - New DASD ioctl to read the 'Sense Path Group ID' data
  - Allows to determine the reservation status of a DASD in relation to the current system
- **Multi-track extension for HPF (kernel 2.6.38)** 6.2 11.2
  - Allows to read from and write to multiple tracks with a single CCW
- **Access to raw ECKD data from Linux (kernel 2.6.38)** 6.2 11.2
  - This item allows to access ECKD disks in raw mode
  - Use the 'dd' command to copy the disk level content of an ECKD disk to a Linux file, and vice versa.
  - Storage array needs to support read-track and write-full-track command.
- **Extended DASD statistics (kernel 3.1)** 6.3
  - Add detailed per-device debugging of DASD I/Os via debugfs
  - Useful to analyze problems in particular for PAV and HPF

Extended DASD statistics

- Start data collection

```
# dasdstat -e dasda 0.0.1234
```

- Reset statistics counters

```
# dasdstat -r dasda
```

- Read summary statistics

```
# dasdstat
statistics data for statistic: 0.0.6527
start time of data collection: Fri Feb 24 16:00:19 CET 2012

1472 dasd I/O requests
with 14896 sectors(512B each)
0 requests used a PAV alias device
0 requests used HPF
 __<4 ___8 __16 __32 __64 _128 _256 _512 __1k __2k __4k __8k _16k _32k _64k 128k
 _256 _512 __1M __2M __4M __8M _16M _32M _64M 128M 256M 512M __1G __2G __4G _>4G
Histogram of sizes (512B secs)
    0    0 1441    8   13    5    2    2    0    1    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
Histogram of I/O times (microseconds)
    0    0    0    0    0    0    1 1160   49   52   61  142    7    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

Linux on System z features – FCP

- **Store I/O and initiate logging (SIOSL) (kernel 2.6.36)** 6.1 11.2
  - Enhance debug capability for FCP attached devices
  - Enables operating system to detect unusual conditions on a FCP channel
- **SAN utilities (kernel 2.6.36, lib-zfcp-hbaapi 2.1)** 6.1 11.2
  - Two new utilities have been added: zfcp_ping and zfcp_show
  - They are useful to discover a storage area network
- **Add NPIV information to symbolic port name (kernel 2.6.39)** 11.2
  - Add the device bus-ID and the network node to the symbolic port name if the NPIV mode is active.

Linux on System z features – Networking

- **Support for assisted VLAN null tagging (kernel 2.6.37)** 6.1 11.2
  - Close a gap between OSA and Linux to process null tagged frames correctly
  - z/OS may sent null-tagged frames to Linux
- **IPv6 support for the qetharp tool (kernel 2.6.38)** 6.2 11.2
  - Extend the qetharp tool to provide IPv6 information in case of a layer 3 setup.
  - This is required for communication with z/OS via HiperSockets using IPv6.
- **New default qeth configuration values (kernel 2.6.39)** 6.2 11.2
  - Receive checksum offload, generic receive offload & number of inbound buffers
- **Add OSA concurrent hardware trap (kernel 3.0)** 6.3 11.2
  - To ease problem determination the qeth driver requests a hardware trace when the device driver or the hardware detect an error
  - Allows to correlate between OSA and Linux traces.

Linux on System z features – s390-tools

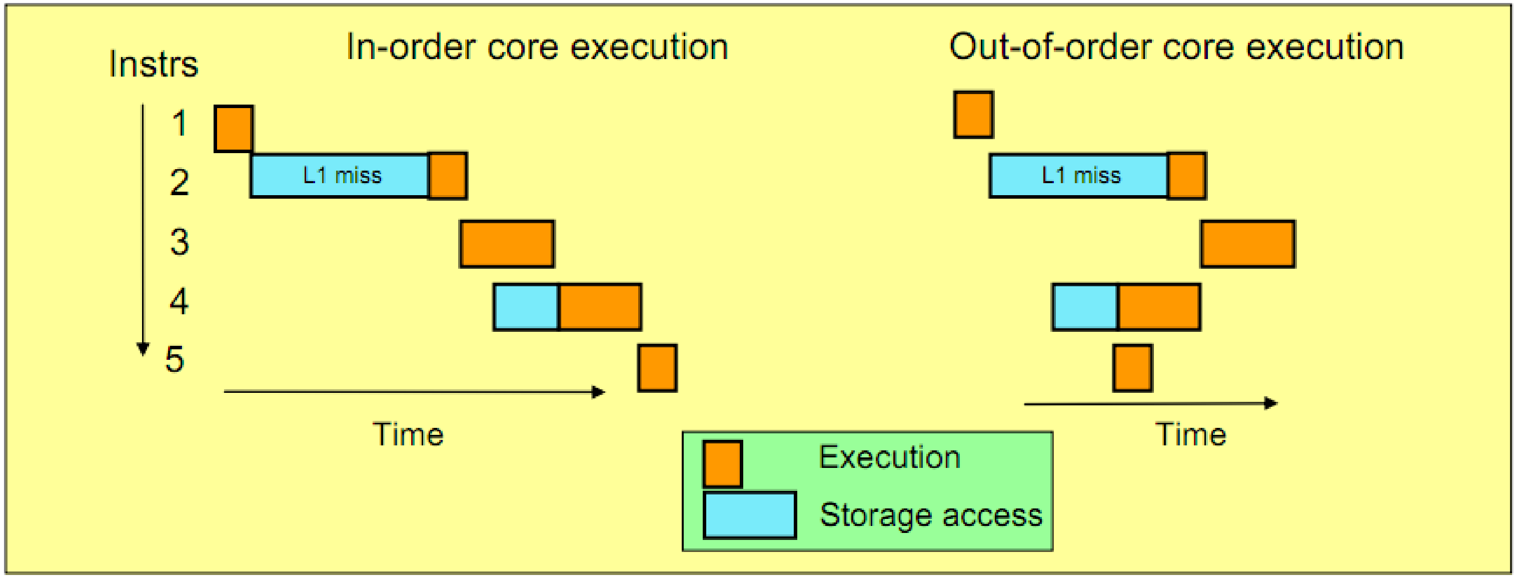
- **Improve memory ballooning with cpuplugd (s390-tools 1.15.0)** 6.2 11.2
  - A number of improvements to the memory balloon part to cpuplugd to make it more useful.
- **reIPL from device-mapper devices (s390-tools 1.12.0)** 6.2 11.2
  - The automatic re-IPL function only works with a physical device
  - Enhance the zipl support for device-mapper devices to provide the name of the physical device if the zipl target is located on a logical device

Linux on System z features – Compiler toolchain

- **zEnterprise 196 exploitation (gcc 4.6)**

6.1 11.2

  - Use option -march=z196 to utilize the new instructions added with z196
  - Use -mtune=z196 to schedule the instruction appropriate for the new out-of-order pipeline of z196
  - Re-compiled code/apps get further performance gains through 110+ new instructions



Linux on System z features – Crypto

- **4096 bit RSA fast path (kernel 2.6.38)** 11.2
  - Make use of 4096 bit RSA acceleration available with Crypto Express 3 GA2 cards.
- **CP ACF exploitation of System z196 (kernel 3.0)** 6.2 11.2
  - Add support for new HW crypto modes:
    cipher feedback mode (CFB), output feedback mode (OFB),
    counter mode (CTR), Galois counter mode (GCM),
    XEX based Tweaked Code Book with Cipher Text Stealing (XTS),
    cipher based message authentication mode (CMAC),
    and counter with cipher block chaining message authentication (CCM)
- **New libica APIs for supported crypto modes (libica 2.1.1)** 6.2 11.2
  - Provide a programmatic way to query for supported crypto ciphers, modes and key sizes.
  - Deliver information whether the cryptographic features are implemented in hardware or in software

LNXHC – Linux Health Checker

- The Linux Health Checker is a command line tool for Linux.
- Its purpose is to identify potential problems before they impact your system's availability or cause outages.
- It collects and compares the active Linux settings and system status for a system with the values provided by health-check authors or defined by you. It produces output in the form of detailed messages, which provide information about potential problems and the suggested actions to take.
- The Linux Health Checker will run on any Linux platform which meets the software requirements. It can be easily extended by writing new health check plug-ins.
- The Linux Health Checker is an open source project sponsored by IBM. It is released under the Eclipse Public License v1.0
- http://lnxhc.sourceforge.net/

Future Linux on System z Technology

Software which has already been developed and integrated into the upstream Linux Kernel - but is **not** yet available in any Enterprise Linux Distribution

Linux on System z features – zEC12 support

- **Transactional execution (kernel 3.7)**
  - Also known as hardware transactional memory
  - CPU features that allows to execute a group of instructions atomically
  - Optimistic execution, if a transaction conflicts a rollback to a saved state is done
- **Storage class memory – Flash Express (kernel 3.7)**
  - Internal Flash Solid State Disk (SSD)
  - Accessed via Extended Asynchronous Data Mover (EADM) sub-channels
  - Up to four pairs of cards with a maximum capacity of 6.4TB
- **Support for Crypto Express 4S cards (kernel 3.7)**
  - New generation of crypto adapters plug-able into the I/O drawer
  - New type 10 which uses a bit field to indicate capabilities of the crypto card
- **Toolchain support (gcc 4.6)**
  - Add new instructions to the compiler
  - Add description of the new pipeline to generate optimal code

System zEC12 features – Transactional Execution

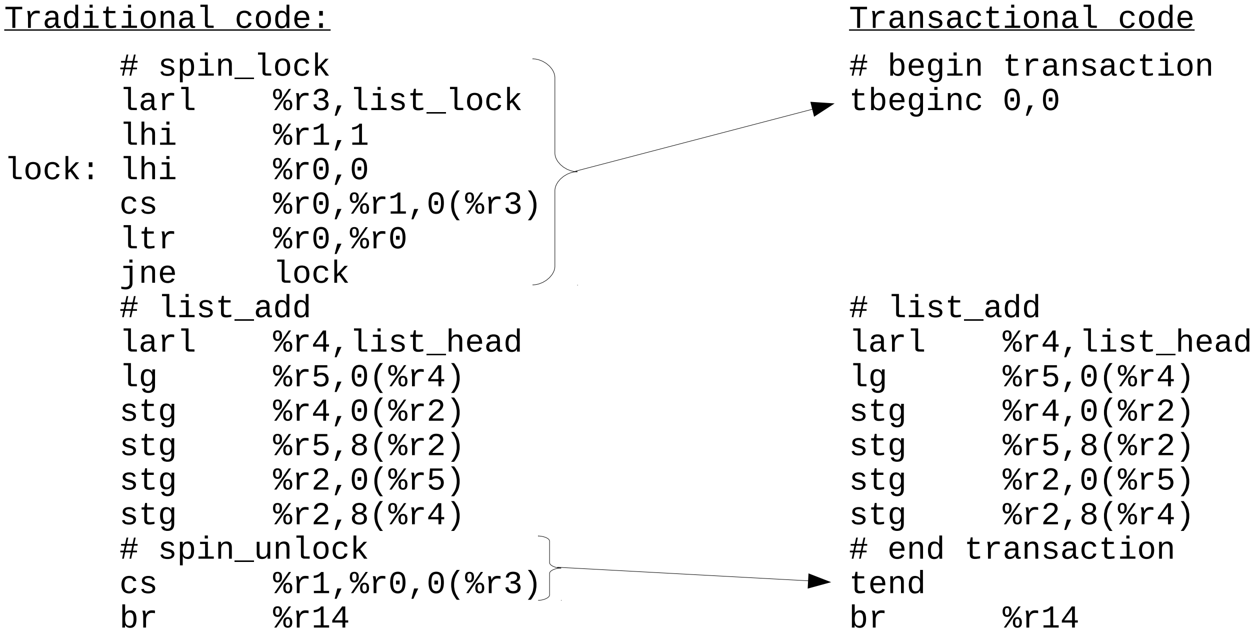
- **Transactional execution is a concurrency mechanism of the CPU comparable to database transactions**
  - Several reads and stores from/to memory logically occur at the same time
  - Improves performance for fine-grained serialization
  - Useful for lock-less data structures and speculative compiler optimizations
- **Two types of transactions: constraint and non-constraint**
- **Conflicting memory accesses will cause the transaction to abort**
  - Transaction abort is rather expensive
  - Constraint transaction will automatically restart
  - Ratio of successful vs. aborted transaction is important for performance
- **Kernel support is required to enable user programs to use transactional execution**
  - Control registers setup
  - Debugging support for additional PER controls via ptrace

System zEC12 features – Transactional Execution

- **Example of a list_add operation**

```
struct spinlock_t list_lock;
struct list_head list_head;
void list_add(struct list_head *new)
{
    spin_lock(&list_lock, 0, 1);
    list_add(new, &list_head);
    spin_unlock(&list_lock, 1, 0);
}
```

Typical pattern:
1) lock, 2) a short operation, 3) unlock



Traditional code:

```
      # spin_lock
      larl    %r3,list_lock
      lhi     %r1,1
lock: lhi     %r0,0
      cs      %r0,%r1,0(%r3)
      ltr     %r0,%r0
      jne     lock
      # list_add
      larl    %r4,list_head
      lg      %r5,0(%r4)
      stg     %r4,0(%r2)
      stg     %r5,8(%r2)
      stg     %r2,0(%r5)
      stg     %r2,8(%r4)
      # spin_unlock
      cs      %r1,%r0,0(%r3)
      br      %r14
```

Transactional code

```
# begin transaction
tbeginc 0,0




# list_add
larl    %r4,list_head
lg      %r5,0(%r4)
stg     %r4,0(%r2)
stg     %r5,8(%r2)
stg     %r2,0(%r5)
stg     %r2,8(%r4)
# end transaction
tend
br      %r14
```

System z kernel features – Core kernel

- **Add support for physical memory > 4TB (kernel 3.3)**
  - Increase the maximum supported memory size from 4TB to 64TB.
  - Memory sizes large than 4TB require a 4-level page table
  - Makes memory accesses by the kernel slightly slower, the kernel will automatically use a 3-level page table for memory sizes <= 4TB
- **Add support for System z hardware counters (kernel 3.4)**
  - Enables hardware counters for Linux running in an LPAR
  - Available counters are: basic counter set, problem-state counter set, crypto-activity counter set, extended counter set with System z10
  - System zEC12 counter set with kernel version 3.7
- **Allow to compare dump system with boot system**
  - With z/VM 6.2 Single-System-Image it is possible to move active Linux instances between different z/VM instances
  - To aid debugging a log of past live-guest-relocations is made available in both the live system and in the dump of a system

System z kernel features – Core kernel

- **Fuzzy Live Dump (kernel 3.5)**
  - Add the capability to generate a dump of a live system.
  - Not all data structures will be consistent but the dump may still be useful.
- **BPF JIT compiler for System z (kernel 3.7)**
  - The Berkeley Packet Filter is an interface and a language definition that allows to pass a filter to the kernel to select network packets to send on a socket
  - The BPF JIT compiler in the kernel translates the interpreted BPF code to System z code.
  - A secondary use of the BFP language is system call filtering.
- **Expose CPU cache topology in sysfs (kernel 3.7)**
  - Add an interface to expose the CPU cache topology to user space.
  - System z only provides information about CPU caches which are private to a CPU, information about shared caches is not exposed.
- **Add page table dumper (kernel 3.7)**
  - Add a sysfs interface to read the current layout of the kernel address space.
  - Useful information for the kernel developer.

System z kernel features – Core kernel / crypto

- **Transparent huge page support (kernel 3.7)**
  - Make the common code transparent huge page support available for Linux on System z.
  - With THP 1MB pages will be used to back normal anonymous memory mappings.
  - Any application will benefit from using huge pages.
  - Not as effective as using the large pages directly, no memory savings for page tables due to huge page splitting.
- **AP adapter resiliency (kernel 3.7)**
  - Improve RAS capabilities of the AP bus and the zcrypt devices.
  - External AP bus configuration changes are now handled correctly.

System z kernel features – Storage FICON / FCP

- **DASD sanity check to detect path connection errors (kernel 3.3)**
  - An incorrect physical connection between host and storage server which is not detected by hardware or microcode can lead to data corruption
  - Add a check in the DASD driver to make sure that each available channel path leads to the same storage server
- **FICON Express8S hardware data router support for FCP (kernel 3.2)**
  - Hardware data router support requires an adapted qdio request format.
  - Improves performance by reducing the path length for data.
- **FCP support for DIF/DIX (kernel 3.2)**
  - End to end data checking (aka data integrity extension) is no longer experimental.
  - Can be used with either direct I/O with a file system that fully supports end-to-end data consistency checking. Currently XFS only.
- **No automatic port rescan on events (kernel 3.7)**
  - The rescan of a zfcp port following a fabric change event can cause high fabric traffic, especially when many Linux images share an FCP channel over multiple subchannels with NPIV enabled. This can lead to errors due to timeouts.
  - Ports are still scanned when the adapter is set online and on manual user triggered writes to the port_rescan sysfs attribute.

System z kernel features – Networking

- **Add support for AF_IUCV HiperSockets transport (kernel 3.2)**
  - Use HiperSockets with completion queues as transport channel for AF_IUCV sockets
- **Allow multiple paths with netiucv between z/VM guests (kernel 3.3)**
  - Adds support for multiple point-to-point netiucv interfaces between guests
  - Speed up netiucv by using parallel IUCV paths.
- **Add query OSA address table support (kernel 3.4)**
  - The new qethqoat command queries the OSA address table and displays physical and logical device information.
  - Viewing the OSA address table is a low level diagnostic requirement.

Linux on System z features – tools

- **Extend lscpu tool and add new chcpu tool (util-linux 2.21)**
  - Improve the lscpu tool to display CPU topology and CPU state
  - Add the new chcpu tool to change CPU state, rescan CPUs and change the CPU dispatching mode (horizontal vs. vertical polarization)
- **SCSI device management tool (> s390-tools 1.14.0)**
  - Implement a tool analog chccwdev which allows to enable/disable a SCSI LUN addressed by HBA/target port/LUN.

System z toolchain

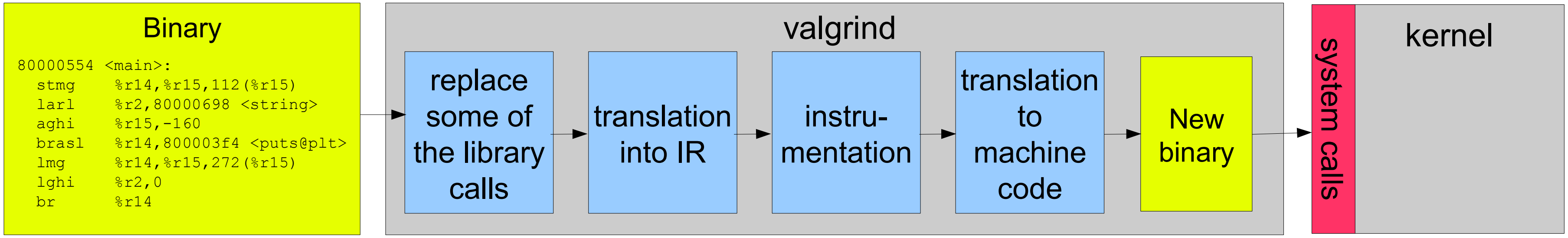
- **64 bit register in 31 bit compat mode (gcc 4.6, glibc 2.11)**
  - Make use of 64 bit registers in 31 bit application running in z/Architecture mode.
  - Allows to use instruction operating on 64 bits, e.g. 64 bit multiplication
  - Needs kernel support for asynchronous signals
- **Exploitation of zEC12 instructions (gcc 4.6)**
  - Use option -march=zEC12 to utilize the new instructions added with zEC12
  - Use -mtune=zEC12 to schedule the instruction appropriate for the pipeline of zEC12
- **ATLAS support (libatlas 3.9.52)**
  - Add support for System z to the “Automatically Tuned Linear Algebra Software”.
  - Improve performance of the library functions for System z.

System z toolchain

- **Oprofile support for hardware sampling introduced with z10 (2.6.39)**
  - Provide CPU measurement data to applications for performance tuning
  - Based on hardware counters and samples built into the CPU
  - Use oprofile to communicate the information to user space programs
- **Oprofile z196 hardware customer mode sampling (kernel 3.3)**
  - Extend the hardware sampling to support z196.
- **Valgrind System z support**
  - Valgrind is a generic framework for creating dynamic analysis tools and can be used for memory debugging, memory leak detection and profiling (e.g. cachegrind)
  - Valgrind is in essence a virtual machine using just-in-time (JIT) compilation techniques
  - Memory debugging is available with Valgrind version 3.7.0

Valgrind System z support

- Several tools, selected with --tool=xxx
  - Memcheck (default): detects memory and data flow problems
  - Cachegrind: cache profiling
  - Massif: heap profiling
  - Helgrind: thread debugging
  - DRD: thread debugging
  - None: no debugging (for valgrind JIT testing)
  - Callgrind: codeflow and profiling
- System z support since version 3.7 (SLES-11-SP2)
- Backports into 3.6 (SLES-10-SP4, RHEL6-U1)



```
80000554 <main>:
  stmg    %r14,%r15,112(%r15)
  larl    %r2,80000698 <string>
  aghi    %r15,-160
  brasl   %r14,800003f4 <puts@plt>
  lmg     %r14,%r15,272(%r15)
  lghi    %r2,0
  br      %r14
```

s390-tools package: what is it?

- s390-tools is a package with a set of user space utilities to be used with the Linux on System z distributions.
  - It is **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 .
- This software package is contained in all major (and IBM supported) enterprise Linux distributions which support s390
  - RedHat Enterprise Linux version 4, 5, and 6
  - SuSE Linux Enterprise Server version 9, 10, and 11
- Website: http://www.ibm.com/developerworks/linux/linux390/s390-tools.html
- Feedback: linux390@de.ibm.com

s390-tools package: the content

chccwdev
chchp
chreipl
chshut
chcrypt
chmem
CHANGE

lscss
lschp
lsdasd
lsluns
lsqeth
lsreipl
lsshut
lstape
lszcrypt
lszfcp
lsmem
DISPLAY

dasdfmt
dasdinfo
dasdstat
dasdview
fdasd
tunedasd
DASD

mon_fsstatd
mon_procd
ziomon
hyptop
MONITOR

ip_watcher
osasnmpd
qetharp
qethconf
NETWORK

tape390_display
tape390_crypt
TAPE

dbginfo
dumpconf
zfcpdump
zfcpdbf
zgetdump
scsi_logging_level
DUMP & DEBUG

vmconvert
vmcp
vmur
cms-fuse
z/VM

cpuplugd
iucvconn
iucvtty
ts-shell
ttyrun
MISC

zipl
BOOT

Kernel news – Common code

- **Linux version 3.2 (2012-01-04)**
  - btfs improvements: faster scrubbing, automatic backup of tree roots
  - ext4: support for bigger block sizes up to 1MB
  - Process bandwidth controller
  - I/O less dirty throttling
  - New architecture: Hexagon

- **Linux version 3.3 (2012-03-18)**
  - Open vSwitch – software multilayer network switch
  - Teaming – better bonding of network interfaces
  - Network transmit queue limits – bufferbloat fighting
  - New architecture: TI C6X

Kernel news – Common code

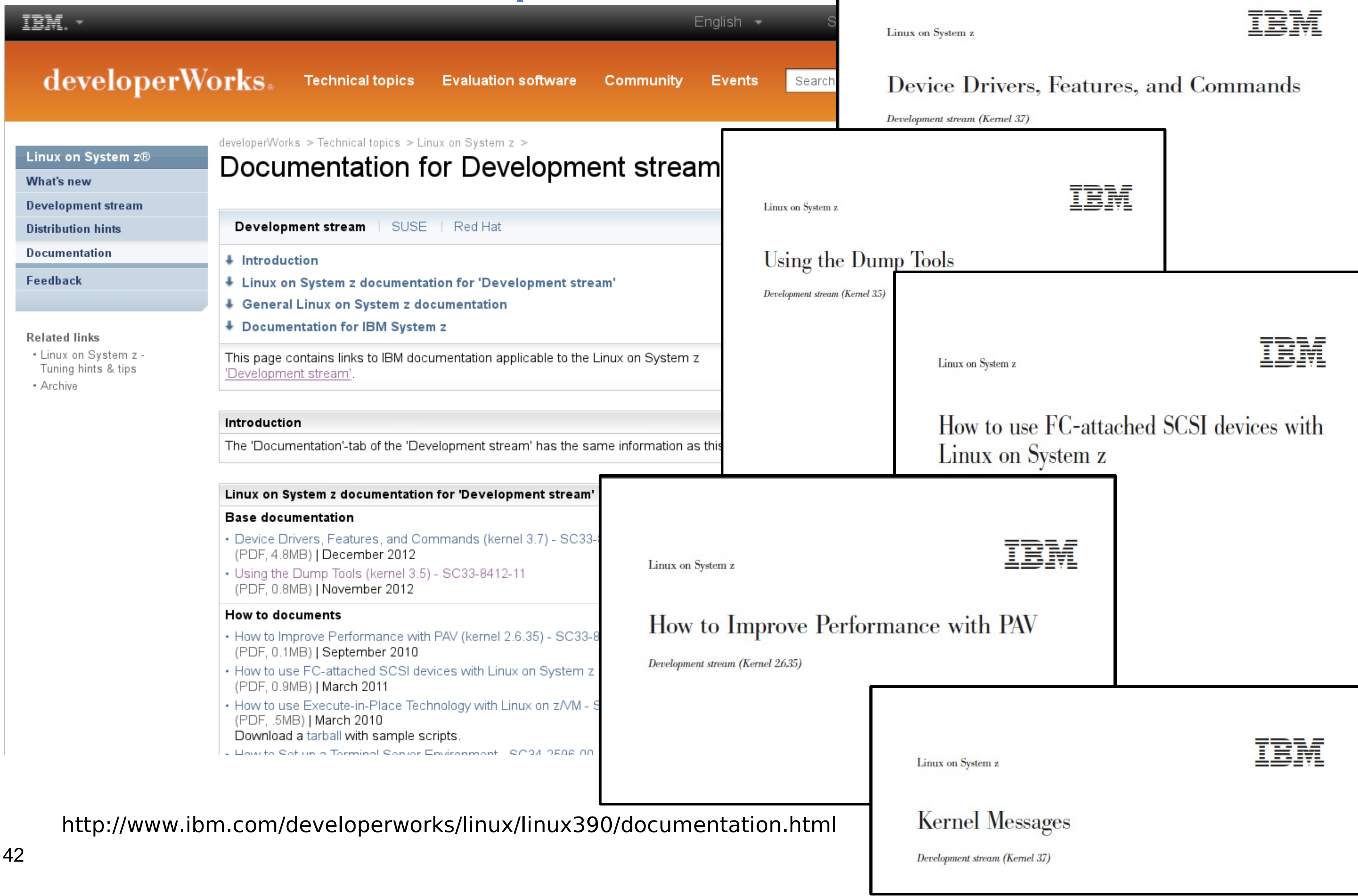
- **Linux version 3.4 (2012-05-20)**
  - Btrfs improvements: better error handling, repair and recovery tools
  - New X32 ABI: 64-bit mode with 32-bit pointers
  - perf tool improvements
  - YAMA security module
  - QNX6 filesystem

- **Linux version 3.5 (2012-07-21)**
  - Filesystem improvements: ext4 metadata checksums, btrfs latency reduction
  - Uprobes: userspace probes
  - Bufferbloat improvements: CoDel queue management
  - TCP: connection repair, early retransmit
  - SCSI over FireWire and USB

Kernel news – Common code

- **Linux version 3.6 (2012-09-30)**
  - Btrfs: subvolume quotas, quota groups, snapshot diff, cross-subvolume file clones
  - TCP Fast Open (client side)
  - Bufferbloat fight: TCP small queues
  - Safe swap over NFS/NBD
- **Linux version 3.7 (2012-12-10)**
  - ARM multi-platform support, 64-bit support
  - Btrfs updates: fsync speedups, hole punching, per-file no-cow support
  - TCP Fast Open (server side)
  - Experimental SMB 2 protocol support

More information: Developer Works



Linux on System z®

- What's new
- Development stream
- Distribution hints
- Documentation
- Feedback

Related links

- Linux on System z - Tuning hints & tips
- Archive

developerWorks > Technical topics > Linux on System z >

Documentation for Development stream

Development stream | SUSE | Red Hat

- Introduction
- Linux on System z documentation for 'Development stream'
- General Linux on System z documentation
- Documentation for IBM System z

This page contains links to IBM documentation applicable to the Linux on System z 'Development stream'.

Introduction

The 'Documentation'-tab of the 'Development stream' has the same information as this

Linux on System z documentation for 'Development stream'

Base documentation

- Device Drivers, Features, and Commands (kernel 3.7) - SC33-
  (PDF, 4.8MB) | December 2012
- Using the Dump Tools (kernel 3.5) - SC33-8412-11
  (PDF, 0.8MB) | November 2012

How to documents

- How to Improve Performance with PAV (kernel 2.6.35) - SC33-8
  (PDF, 0.1MB) | September 2010
- How to use FC-attached SCSI devices with Linux on System z
  (PDF, 0.9MB) | March 2011
- How to use Execute-in-Place Technology with Linux on z/VM - S
  (PDF, .5MB) | March 2010
  Download a tarball with sample scripts.
- How to Set up a Terminal Server Environment - SC34-2596-00

Linux on System z

Device Drivers, Features, and Commands

Development stream (Kernel 3.7)

Linux on System z

Using the Dump Tools

Development stream (Kernel 3.5)

Linux on System z

How to use FC-attached SCSI devices with Linux on System z

Linux on System z

How to Improve Performance with PAV

Development stream (Kernel 2.6.35)

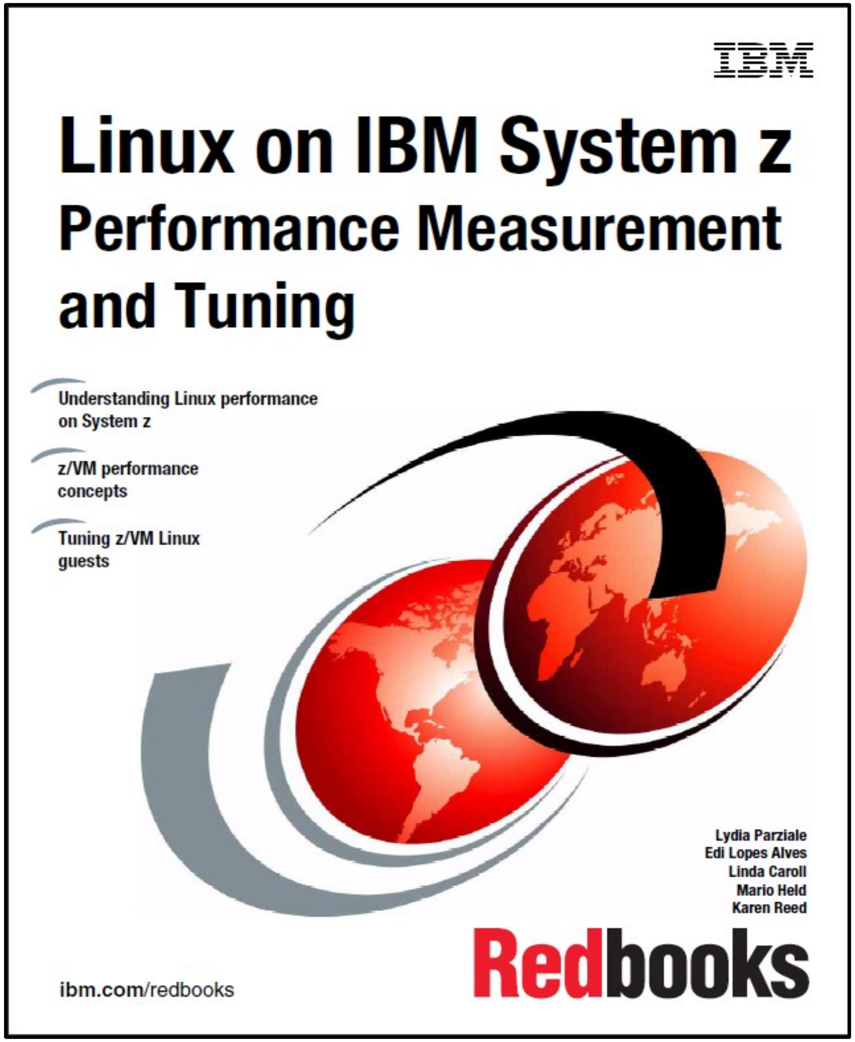
Linux on System z

Kernel Messages

Development stream (Kernel 3.7)

http://www.ibm.com/developerworks/linux/linux390/documentation.html

New Redbooks



Linux on IBM System z Performance Measurement and Tuning

Understanding Linux performance on System z

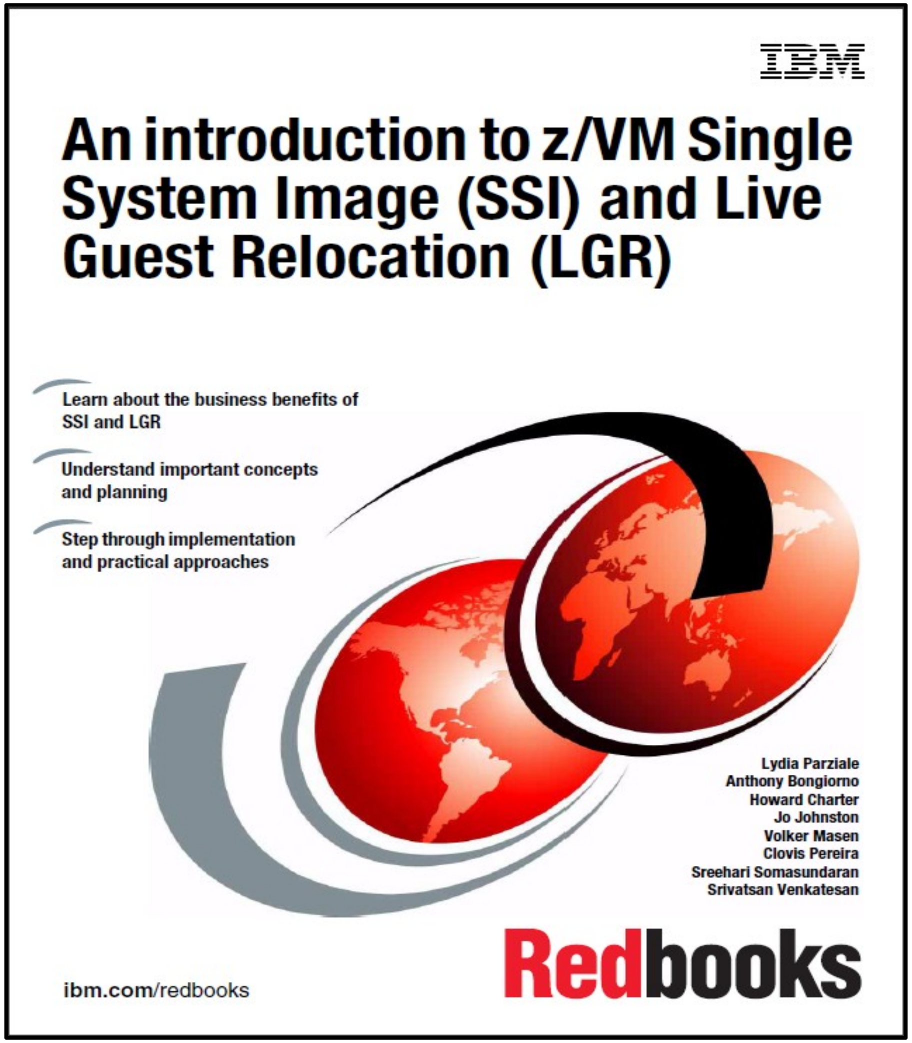
z/VM performance concepts

Tuning z/VM Linux guests

Lydia Parziale
Edi Lopes Alves
Linda Caroll
Mario Held
Karen Reed

ibm.com/redbooks

Redbooks



An introduction to z/VM Single System Image (SSI) and Live Guest Relocation (LGR)

Learn about the business benefits of SSI and LGR

Understand important concepts and planning

Step through implementation and practical approaches

Lydia Parziale
Anthony Bongiorno
Howard Charter
Jo Johnston
Volker Masen
Clovis Pereira
Sreehari Somasundaran
Srivatsan Venkatesan

ibm.com/redbooks

Redbooks

Visit http://www.redbooks.ibm.com

Questions?

Martin Schwidefsky

Linux on System z
Development

Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 (0)7031-16-2247
schwidefsky@de.ibm.com

Please Evaluate!