



What's New in Linux on IBM Z

Martin Schwidefsky



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, z Systems, 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.



Trademarks & Disclaimer #2

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

Notice Regarding Specialty Engines

Any information contained in this document regarding Specialty Engines (“SEs”) and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the “Authorized Use Table for IBM Machines” provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html (“AUT”).

No other workload processing is authorized for execution on an SE.

IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.



Linux on IBM z distributions

What is available today



Linux on IBM z distributions

- **SUSE Linux Enterprise Server 10**

- 07/2006 SLES10 GA: Kernel 2.6.16, GCC 4.1.0
- 04/2011 SLES10 SP4; **EOS 31 Jul. 2013; LTSS: 30 Jul. 2016**

- **SUSE Linux Enterprise Server 11**

- 03/2009 SLES11 GA: Kernel 2.6.27, GCC 4.3.3
- 07/2015 SLES11 SP4: Kernel 3.0, GCC 4.3.4; EOS 31 Mar. 2019; LTSS: 31 Mar. 2022

- **SUSE Linux Enterprise Server 12**

- 10/2014 SLES12 GA: Kernel 3.12, GCC 4.8
- 09/2017 SLES12 SP3: Kernel 4.4, GCC 4.8
- Last SP: EOS 31 Oct. 2024; LTSS: 31 Oct. 2027



Linux on IBM z distributions

- **Red Hat Enterprise Linux AS 5**
 - 03/2007 RHEL5 GA: Kernel 2.6.18, GCC 4.1.0
 - 09/2014 RHEL5 Update 11; EOS 31 Mar. 2017; ELS: 30 Nov. 2020
- **Red Hat Enterprise Linux AS 6**
 - 11/2010 RHEL6 GA: Kernel 2.6.32, GCC 4.4.0
 - 03/2017 RHEL6 Update 9; EOS 30 Nov. 2020; ELS: tbd
- **Red Hat Enterprise Linux AS 7**
 - 06/2014 RHEL7 GA: Kernel 3.10, GCC 4.8
 - 08/2017 RHEL7 Update 4; EOS 30 Jun. 2024; ELS: tbd



Linux on IBM z distributions


- **Ubuntu 16.04 (Xenial Xerus)**

- Canonical and IBM announced an Ubuntu based distribution on LinuxCon 2015 in Seattle
- 04/2016 Ubuntu 16.04 GA: Kernel 4.4, GCC 5.3.0+ LTS-Release
- 10/2016 Ubuntu 16.10 GA: Kernel 4.8, GCC 6.2.0+
- 04/2017 Ubuntu 17.04 GA: Kernel 4.10, GCC 6.3.0+
- 10/2017 Ubuntu 17.10 GA: Kernel 4.13, GCC 7.2.0+
- Lifecycle:
 - Regular releases every 6 months and supported for 9 months
 - LTS releases every 2 years and supported for 5 years
 - LTS enablement stack will provide newer kernels within LTS releases
 - <http://www.ubuntu.com/info/release-end-of-life>



Supported Linux Distributions

| Distribution | LinuxONE Emperor II | LinuxONE Emperor | LinuxONE Rockhopper | | | |
|--------------|------------------------|---------------------|------------------------|----------------------------------|--------------------------------|-----------------------------|
| | z14 | z13 | z13s | zEnterprise - zBC12 and zEC12 | zEnterprise - z114 and z196 | System z10 and System z9 |
| RHEL 7 | ✓ (1) | ✓ (4) | ✓ (4) | ✓ (7) | ✓ (7) | ✗ |
| RHEL 6 | ✓ (**) | ✓ (4) | ✓ (4) | ✓ (8) | ✓ | ✓ |
| RHEL 5 | ✗ | ✓ (4) | ✗ | ✓ (9) | ✓ | ✓ |
| RHEL 4 (*) | ✗ | ✗ | ✗ | ✗ | ✓ (12) | ✓ |
| SLES 12 | ✓ (2) | ✓ (5) | ✓ (5) | ✓ | ✓ | ✗ |
| SLES 11 | ✓ (2) | ✓ (5) | ✓ (5) | ✓ (10) | ✓ | ✓ |
| SLES 10 (*) | ✗ | ✗ | ✗ | ✓ (11) | ✓ | ✓ |
| SLES 9 (*) | ✗ | ✗ | ✗ | ✗ | ✓ (13) | ✓ |
| Ubuntu 16.04 | ✓ (3) | ✓ (6) | ✓ (6) | ✓ (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. Please check with your service provider which kernel-levels are currently in support.

See www.ibm.com/systems/z/os/linux/resources/testedplatforms.html for latest updates and details.



Linux on IBM z distributions

- Please check the tested-platforms web link for minimum required kernel levels
- Notes about IBM z14
 - RHEL5 is **not** supported on IBM z14, SLES10 has not been supported on IBM z13 already
 - Tested platforms current has the following footnote on z14:
“IBM is working with the Linux partner to support selected levels of the distribution on z14.”
 - RHEL6 and SLES11 required at least one small patch
 - RHEL7 and SLES12 run fine on z14, but do not have “z14” in the ELF platform name



Linux support for IBM z14

Machine support code for IBM z14, partially already upstream with a few more features under development.



IBM z14 Support

- **Toleration for Crypto Express 6 cards (kernel 4.10)**



- Allow to use the new crypto hardware in CEX5 compat mode

- **Report new vector facilities (kernel 4.11)**



- Add two new features flags in /proc/cpuinfo: “vxd” for the Vector-Decimal Facility and “vxe” for the Vector-Enhancement Facility 1
- No additional enablement is required, if vector instruction are available the two new facilities are enabled as well

- **Instruction execution protection (kernel 4.11)**



- Also know as non-executable mappings or short “noexec”
- New bits in the segment and page tables can be used to forbid code execution for a 1M segment or a 4K page
- The PROT_EXEC flag of mmap / mprotect already provides the information which memory regions contains instruction vs. data
- The presence of the GNU_STACK program header without the execute flag makes all memory mappings with PROT_EXEC==0 to be non-executable



IBM z14 Support: Instruction Execution Protection



- Example: memory map of a simple program via 'cat /proc/<pid>/maps'

```

00001000000-00001001000 r-xp 00000000 5e:0d 391688 /usr/src/hello
00001001000-00001002000 r--p 00000000 5e:0d 391688 /usr/src/hello
00001002000-00001003000 rw-p 00001000 5e:0d 391688 /usr/src/hello
000159a2000-000159c3000 rw-p 00000000 00:00 0 [heap]
3ff81600000-3ff817b8000 r-xp 00000000 5e:01 1183055 /usr/lib64/libc-2.24.so
3ff817b8000-3ff817bc000 r--p 001b7000 5e:01 1183055 /usr/lib64/libc-2.24.so
3ff817bc000-3ff817be000 rw-p 001bb000 5e:01 1183055 /usr/lib64/libc-2.24.so
3ff817be000-3ff817c2000 rw-p 00000000 00:00 0
3ff81880000-3ff818a7000 r-xp 00000000 5e:01 1179503 /usr/lib64/ld-2.24.so
3ff818a8000-3ff818a9000 r--p 00027000 5e:01 1179503 /usr/lib64/ld-2.24.so
3ff818a9000-3ff818aa000 rw-p 00028000 5e:01 1179503 /usr/lib64/ld-2.24.so
3ff818aa000-3ff818ab000 rw-p 00000000 00:00 0
3ff818fa000-3ff818fe000 rw-p 00000000 00:00 0
3ff818fe000-3ff81900000 r-xp 00000000 00:00 0 [vdso]
3fffd2df000-3fffd300000 rw-p 00000000 00:00 0 [stack]

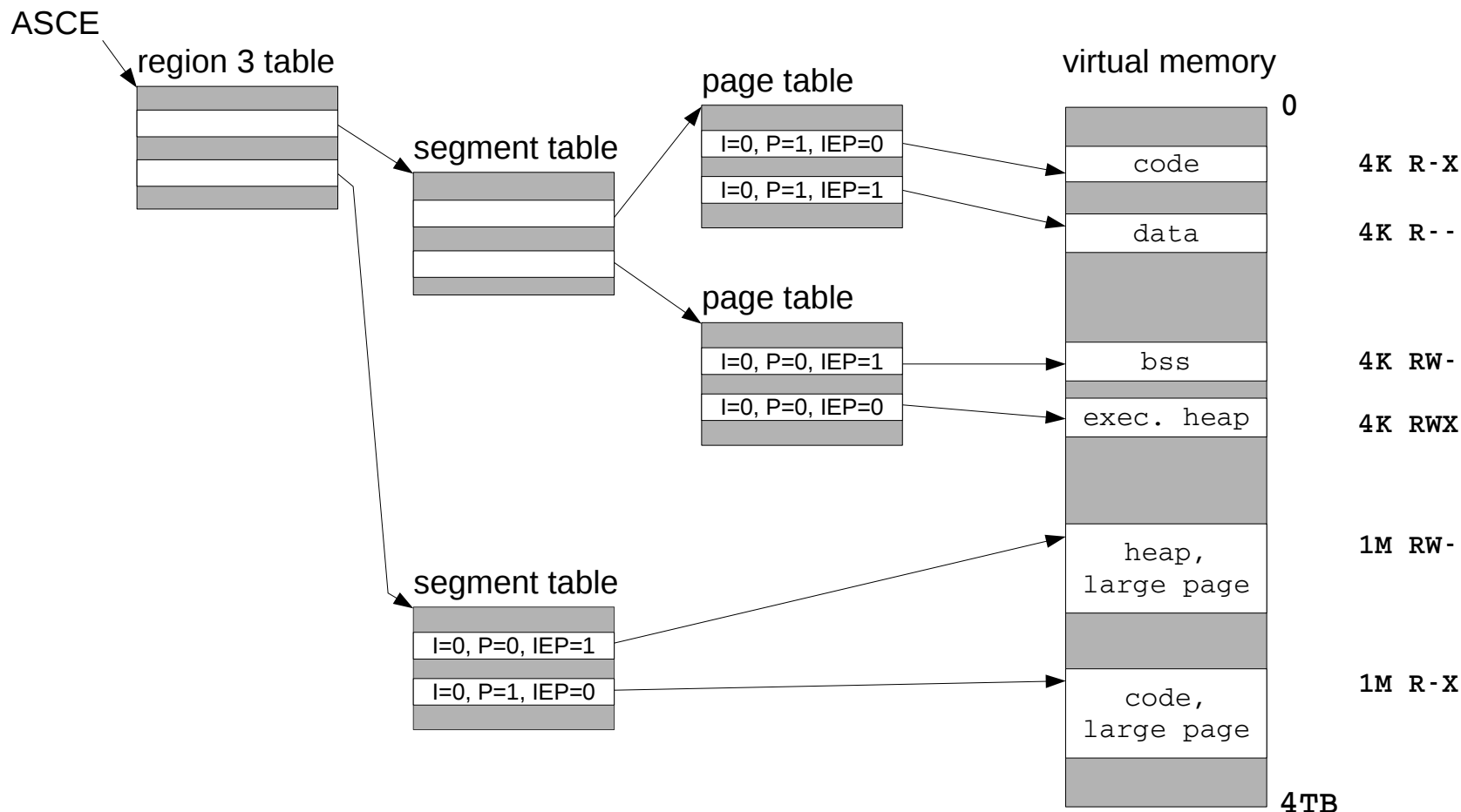
```

r: readable
 w: writable
 x: executable
 s: shared
 p: private

< z14: read-implies-exec
 >= z14: exec-requires-read, exec can
 be disabled independently



IBM z14 Support: Instruction Execution Protection

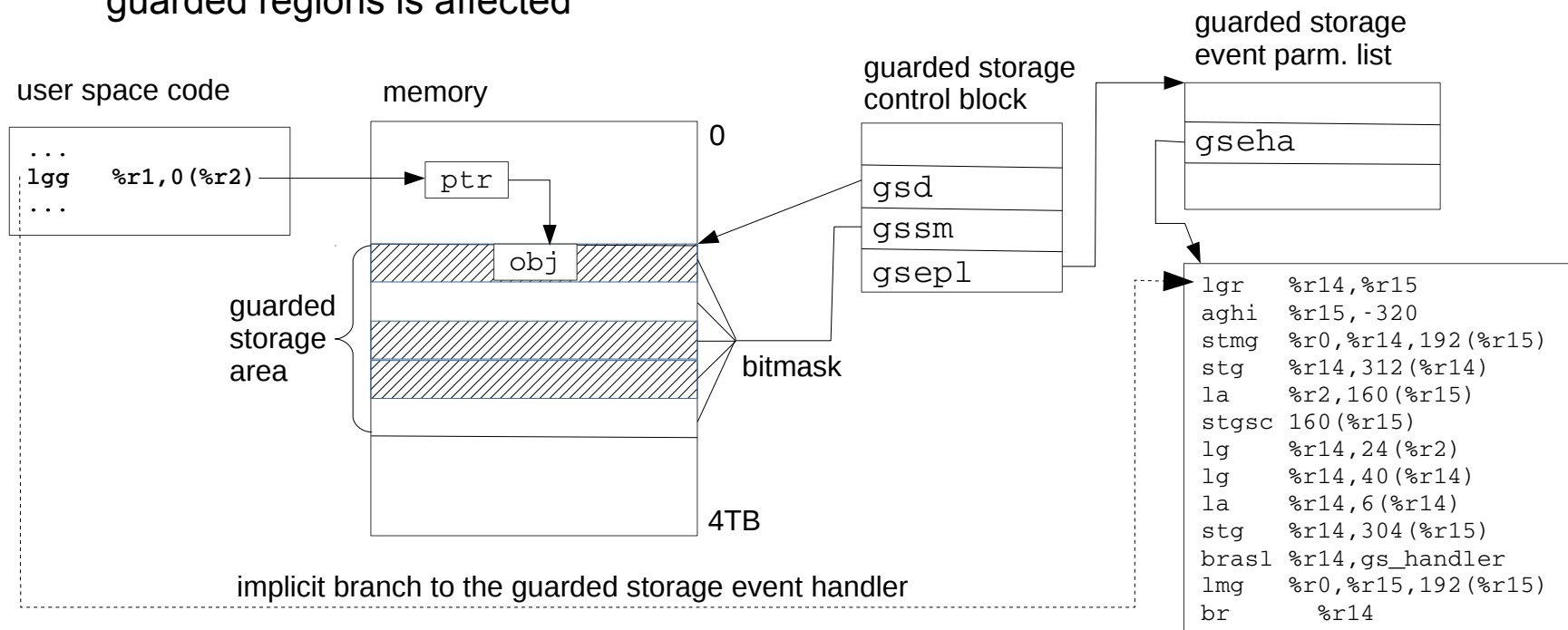


IBM z14 Support: Pauseless Garbage Collection

• Support for the Guarded Storage Facility (kernel 4.12)



- Designed to improve the performance of Java while garbage collection is active
- Up to 64 regions of memory can be marked as guarded
- Reading a pointer with the new LGG or LLGFSG instruction will do a range check on the loaded value and automatically invoke a user space handler if one of the guarded regions is affected



IBM z14 Support: Base Kernel

- **TOD-Clock Extensions for Multiple Epochs (> kernel 4.12)**



- On September 17, 2042 at 23:53:57.370496 TAI the 64-bit TOD clock will overflow
- The extended TOD clock format has 8 additional bits, the epoch index
- Make Linux work with a wrapped 64-bit TOD clock and clock comparators

- **Single-Increment-Assignment Control for memory hotplug (> kernel 4.12)**



- Speed up the Attach-Storage-Element SCLP request
- Improves operation time for some memory hotplug operations

- **Optimized spinlocks with NIAI (> kernel 4.12)**



- Use the new sub codes 4, 7 and 8 of the NIAI instruction to reduce cache line traffic due to congested spinlocks



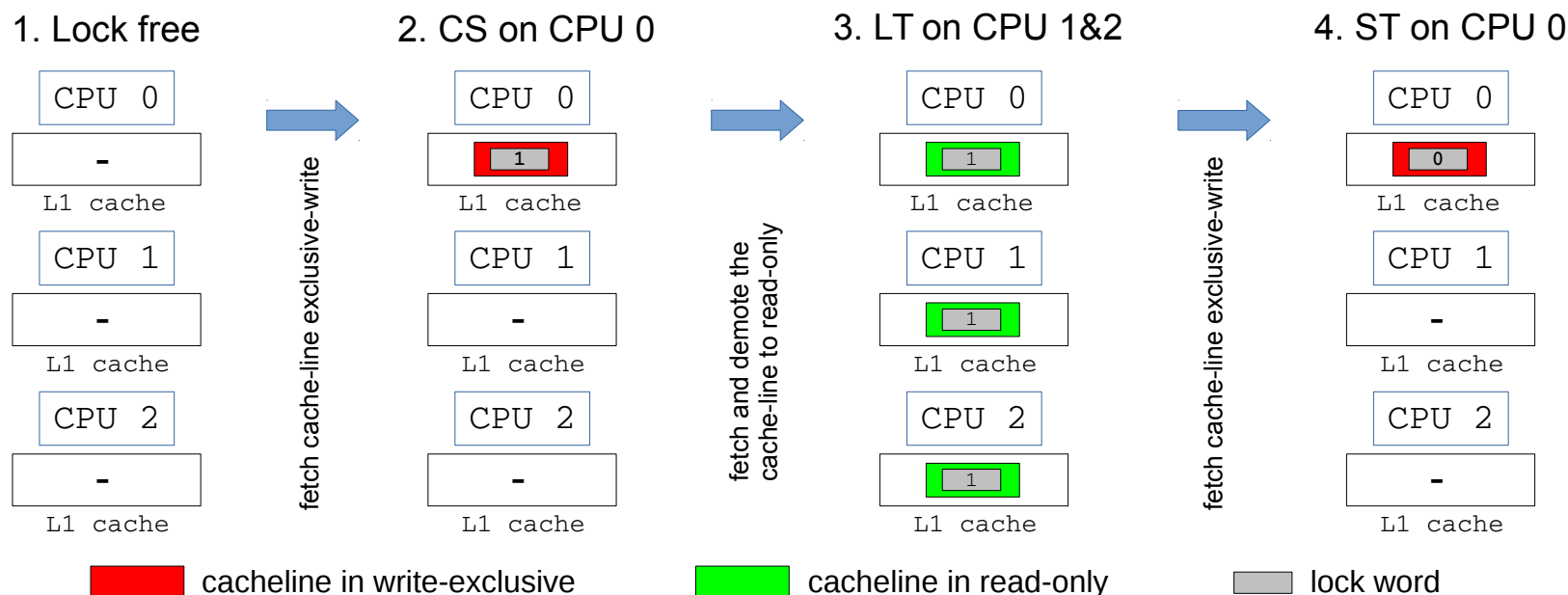
IBM z14 Support: Optimized spinlocks with NIAI (simplified)

- Traditional spinlock implementation
 - load-and-test to check for availability
 - compare-and-swap to acquire the lock
 - store of 0 to free the lock

```

larl  %r2,<some_lock>
lhi   %r1,1
# spinlock loop
loop: LT   %r0,0(%r2)      # load and test
      jnz  loop           # not free → loop
      CS   %r0,%r1,0(%r2)  # lock operation
      jnz  loop           # no success → loop
      # locked section
...
# unlock operation (%r0 contains 0)
ST    %r0,0(%r2)          # clear lock word

```



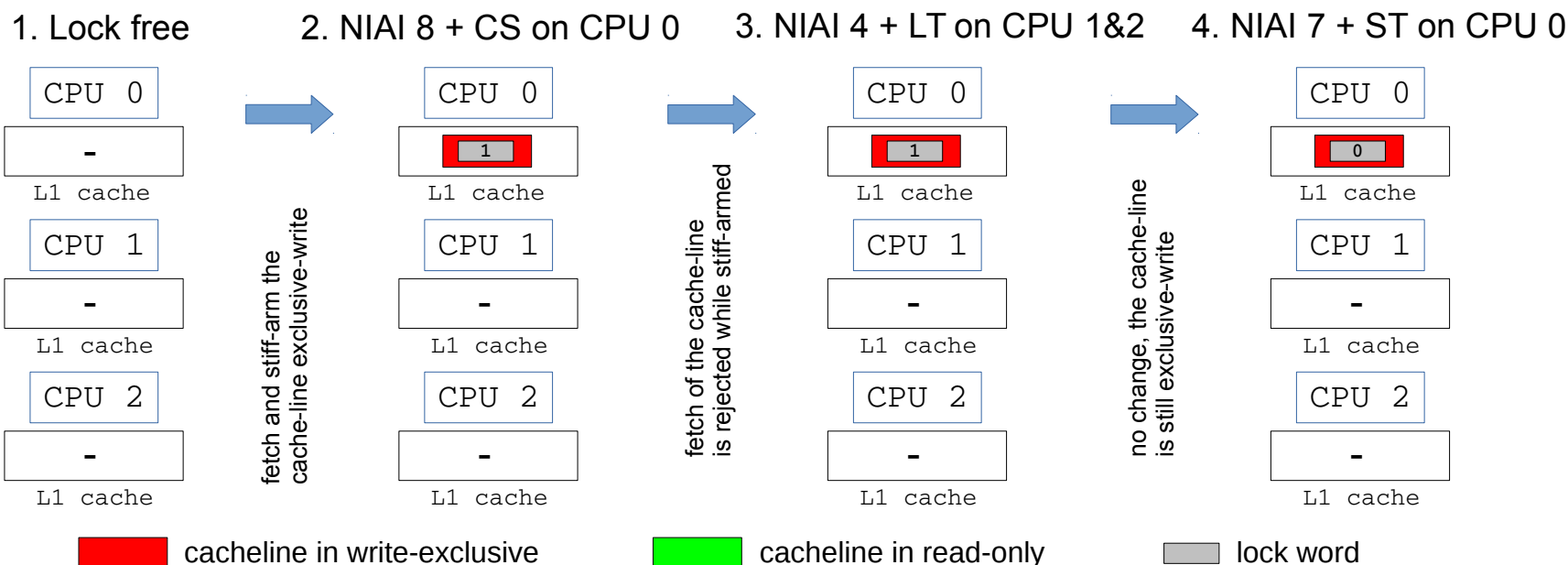
IBM z14 Support: Optimized spinlocks with NIAI (simplified)

- spinlock implementation with NIAI
 - no-speculative-load (NIAI 4) + load-and-test to test for availability
 - hold cache-line (NIAI 8) + compare-and-swap to acquire the lock
 - release cache-line (NIAI 7) + store of 0 to free the lock

```

larl  %r2,<some_lock>
lhi   %r1,1
# spinlock loop
loop: NIAI  4,0
      LT    %r0,0(%r2)      # load and test
      jnz   loop            # not free → loop
      NIAI  8,0             # pin cacheline
      CS    %r0,%r1,0(%r2)  # lock operation
      jnz   loop            # no success → loop
      # locked section
      ...
      # unlock operation (%r0 contains 0)
      NIAI  7,0             # release cacheline
      ST    %r0,0(%r2)      # clear lock word

```



IBM z14 Support: Crypto

- **True random number generator (kernel 4.12)**



- The MSA-7 CPACF extension provides a new function for true random numbers
- Add a hwrng for user space and an arch random function for in-kernel use
- A “dd if=/dev/trng of=/dev/zero bs=1M count=1” returns ~500 KB per second

- **GCM enhancements (kernel, libica, openssl)**



- Exploit the new CPACF instruction for aes-gcm
- Useful for a variety of protocols, e.g. IETF IPsec standard (OpenVPN), IEEE 802.11ad (WiGig), or Fibre Channel Security Protocol (FC-SP)

- **SHA3 enhancement (libica)**



- Exploit the new CPACF instruction for the Secure Hash Algorithm 3
- NIST standard released on August 5, 2015

- **CEX6S and CCA 6.0 support**



- New CCA function will be available with z14 and CEX6S



Future Linux on IBM z Systems Technology

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



Shared Memory Communication over RDMA (SMC-R) and RoCE

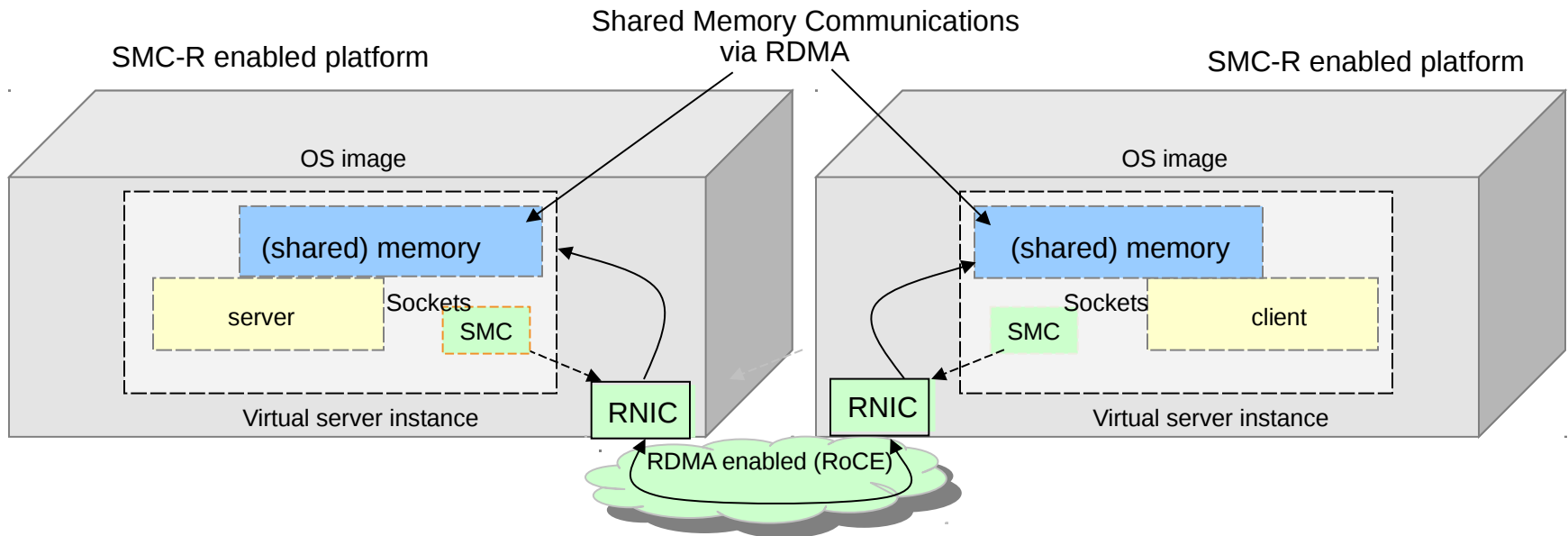
- Shared Memory Communications over RDMA (SMC-R) is a protocol that allows applications to exploit RDMA (RoCE) with the socket interface
- The Linux support for SMC-R uses a new address family **AF_SMC**
 - The addressing scheme is the same as TCP, to “port” an application to SMC-R simply replace AF_INET with AF_SMC:

```
tcp_socket = socket(AF_INET, SOCK_STREAM, 0);  
by  
tcp_socket = socket(AF_SMC, SOCK_STREAM, 0);
```

- Alternatively a preload library available in package SMC Tools at <https://ibm.biz/BdiZ5m> can be used to intercept the socket call
 - Automatic fallback to AF_TCP if the connection could not be established via SMC
- A first version of the Linux code is now upstream with kernel 4.11-rc1
 - The Linux variant is currently **incompatible** with the z/OS version
 - More work on both the Linux and the z/OS side is required to connect Linux to z/OS via SMC-R



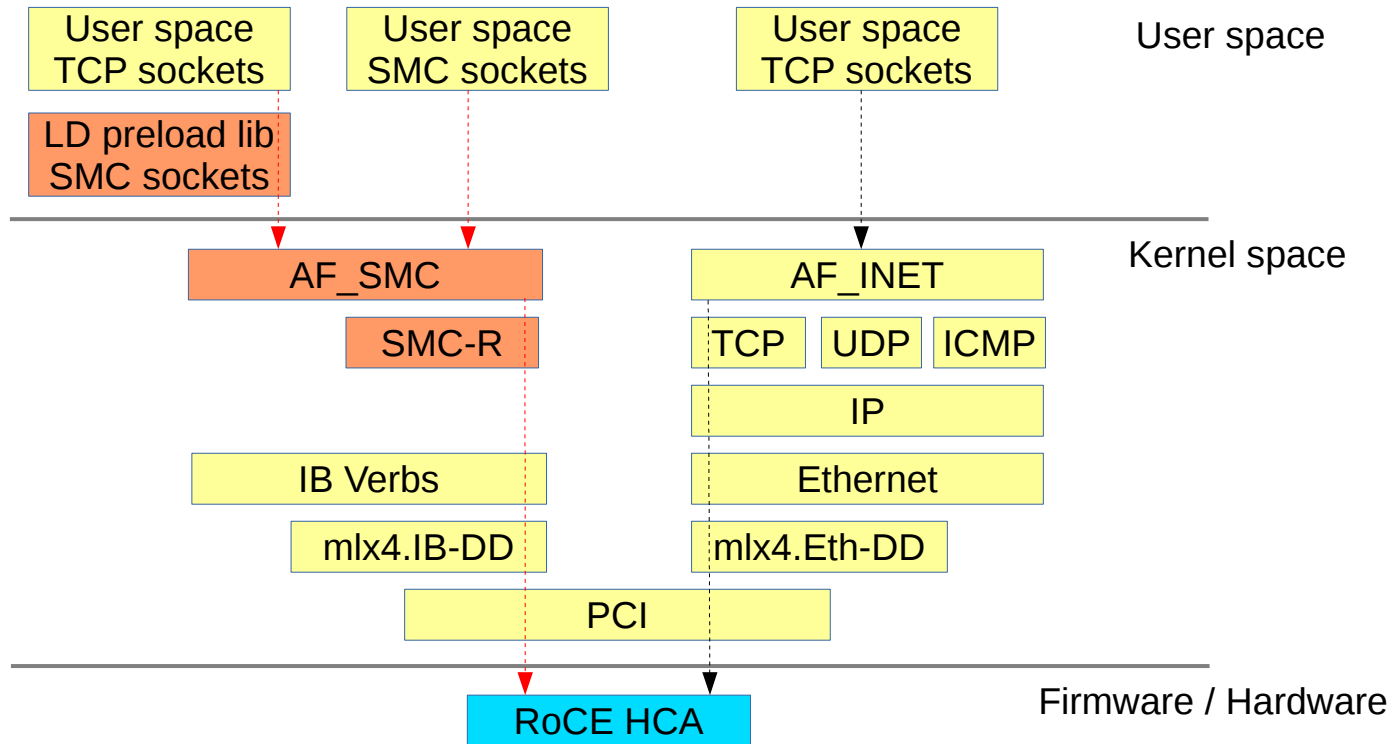
SMC-R concept / overview



RDMA technology provides the capability to allow hosts to logically share memory. The SMC-R protocol defines a means to exploit the shared memory for communications - transparent to the applications!



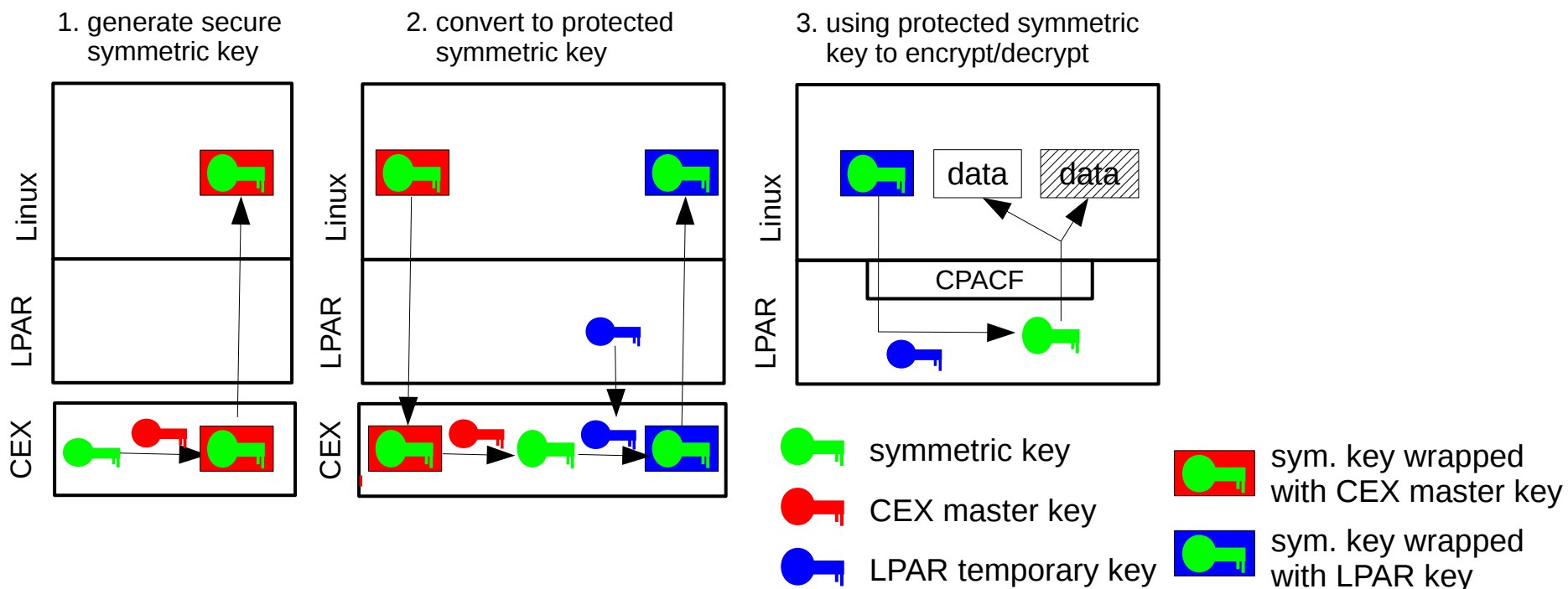
Linux structure for SMC-R



Kernel features – crypto support

• Protected key encryption for dm-crypt (kernel 4.11)

- Consists of the protected key AES module and the secure key API module
- Allows to encrypt block devices without a clear text key anywhere in memory
- Userspace tooling for LUKS1 / LUKS2 needs more work, plain cryptsetup works



Kernel features: PCI improvements

- **PCI error reporting interface (kernel v4.9)**



- Provide a sysfs interface to allow user space programs to trigger a deconfigure-and-repair action for a specific PCI function














- **PCI I/O TLB flush enhancement (kernel v4.10)**



- Reduce the number of RPCIT instructions in case the hypervisor does not announce that RPCIT can be omitted for invalid -> valid translation-table entry updates



Kernel features - miscellaneous

- **Scatter-gather for AF_IUCV sockets (kernel 4.8)** 
 - Avoid large continuous kernel buffer allocations for AF_IUCV under z/VM
- **Show dynamic and static CPU speed in /proc/cpuinfo (kernel 4.8)**   
 - Reports the static and dynamic MHz rating of each CPU
- **Add leap seconds to initial system time (kernel 4.8)**   
 - The current number of leap seconds is a configuration setting of the local machine
 - If the leap seconds have been set correctly they must be subtracted from the TOD clock to determine UTC
- **Performance enhancement for RAID6 gen/xor (kernel 4.9)**   
 - Speed up the RAID6 syndrome and xor functionsq
- **5 level page tables (kernel 4.11 / kernel 4.13)**   
 - For x86 machines support for five level of page tables has been introduced with 4.11
 - The z Systems support is planned for kernel version 4.13
 - The user space address limit for z Systems will be 16EB-4KB



Kernel features - miscellaneous

- **IBM z13 specific CPU-MF counter event names (kernel 4.12)**



- Add the model specific counter event names of the CPU-Measurement Facility for the IBM z13 machine
- Allows to use symbolic names instead of the raw event names 'r[0-9a-z]*'
- Use 'lscpumf -C' for a complete list

- **IBM z13 Multi-Threading CPU-MF counter set (kernel 4.12)**



- Add support for the MT-diagnostic counter set introduced with IBM z13
- Provides access to the counters `MT_DIAG_CYCLES_ONE_THR_ACTIVE` and `MT_DIAG_CYCLES_TWO_THR_ACTIVE`

- **Live patch support (kernel 4.12)**



- Add the architecture backend for z Systems for live patching
- Provides the basis for the kGraft and kpatch solutions, both allow to update a running kernel with critical patches without a downtime



Linux common code enablements

- **KCOV support (kernel v4.8)**



- Aka “Kernel coverage information”
- Exposes kernel code coverage information in a form suitable for coverage-guided fuzzing (randomized testing).

- **UBSAN sanitizer (kernel v4.9)**



- Aka “Undefined behaviour sanity checker”
- Uses compile-time instrumentation to detect undefined behaviours at runtime.

- **CMA support (kernel v4.10)**



- Aka “Contiguous Memory Allocator”
- Allows subsystems to allocate big physically-contiguous blocks of memory.



Questions?

