

GS11 - Linux on IBM z Systems News

Leipzig, 25. October 2016

10th European GSE/IBM Technical University
for z/VSE, z/VM, KVM and Linux on IBM z Systems

*Dr. Manfred Gnirss
IBM Client Center
Böblingen, Germany*



October 24th – 26th, 2016
The Westin Hotel in Leipzig, Germany

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 Vecteded 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.

Acknowledgement

My very best thanks belong to

- Martin Schwidefsky
- Stefan Raspl
- Michael Holzheu

and to all others who contributed to this presentation

Agenda

- Linux on IBM z Systems introduction
- Linux on IBM z Systems distributions
- Current Linux on IBM z Systems Technology
- Future Linux on IBM z Systems Technology

Linux on IBM z Systems introduction

Interesting facts and numbers

Fun facts around Linux on IBM z Systems

- How many git commits are there in the main Linux repository up to v4.6

Fun facts around Linux on IBM z Systems

- How many git commits are there in the main Linux repository up to v4.6
590166 commits (547829 without merge commits, %7 merge commits)
- How many of these git commits are s390 related?

Fun facts around Linux on IBM z Systems

- How many git commits are there in the main Linux repository up to v4.6
590166 commits (547829 without merge commits, %7 merge commits)
- How many of these git commits are s390 related?
~5781 commits (~1%)
- What is the maximum of lines added by a single git commit up to v4.6?

Fun facts around Linux on IBM z Systems

- How many git commits are there in the main Linux repository up to v4.6
590166 commits (547829 without merge commits, %7 merge commits)
- How many of these git commits are s390 related?
~5781 commits (~1%)
- What is the maximum of lines added by a single git commit up to v4.6?
git commit d7e09d0397e84eef “staging: add Lustre file system client support”, 258994 insertions(+)
- What is the maximum of lines added by a single git commit for s390?

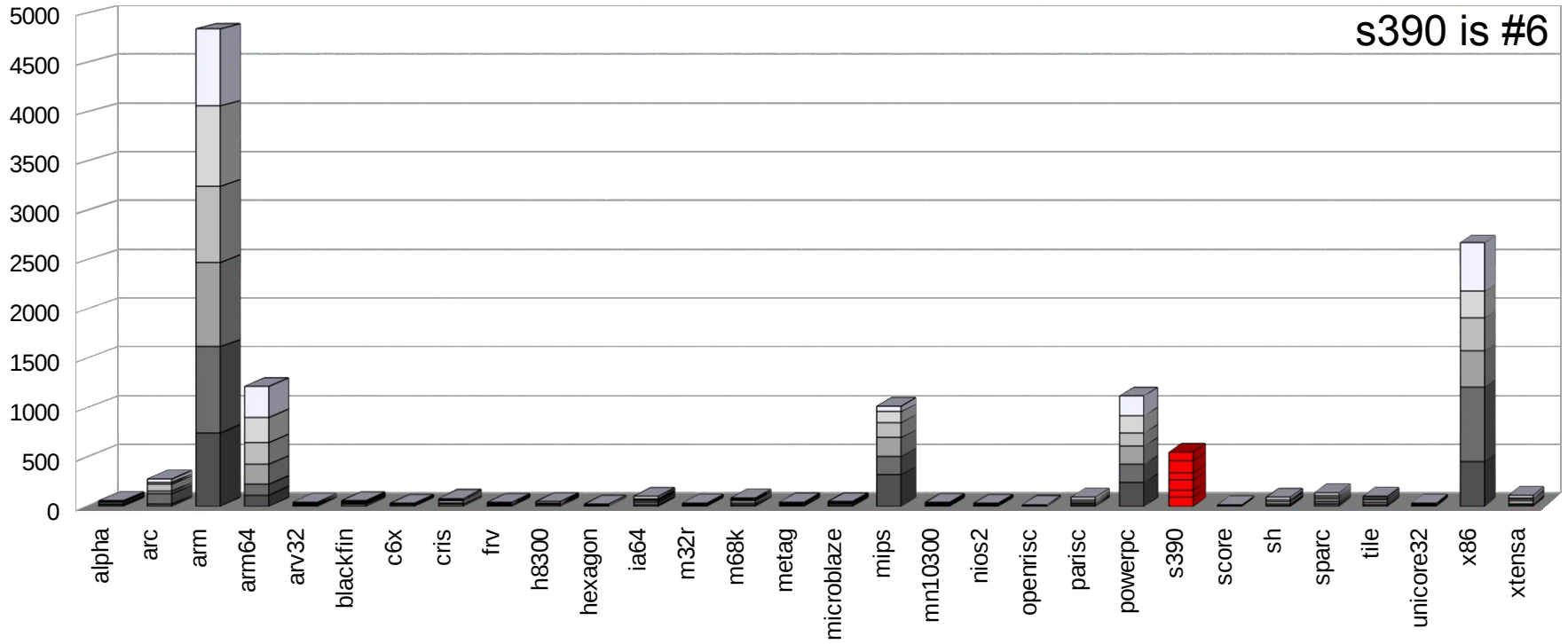
Fun facts around Linux on IBM z Systems

- How many git commits are there in the main Linux repository up to v4.6
590166 commits (547829 without merge commits, %7 merge commits)
- How many of these git commits are s390 related?
~5781 commits (~1%)
- What is the maximum of lines added by a single git commit up to v4.6?
git commit d7e09d0397e84eef “staging: add Lustre file system client support”, 258994 insertions(+)
- What is the maximum of lines added by a single git commit for s390?
git commit 4a71df50047f0db6 “new qeth device driver”, 13498 insertions(+)
- How many individual developers contributes to Linux in 2015 ?

Fun facts around Linux on IBM z Systems

- How many git commits are there in the main Linux repository up to v4.6
590166 commits (547829 without merge commits, %7 merge commits)
- How many of these git commits are s390 related?
~5781 commits (~1%)
- What is the maximum of lines added by a single git commit up to v4.6?
git commit d7e09d0397e84eef “staging: add Lustre file system client support”, 258994 insertions(+)
- What is the maximum of lines added by a single git commit for s390?
git commit 4a71df50047f0db6 “new qeth device driver”, 13498 insertions(+)
- How many individual developers contributes to Linux in 2015 ?
4022 individuals, 1324 with a single commit, 57 contributed directly to s390
(81 contributors including tree wide changes, typo fixes, janitor patches, ...)

Git commits per architecture in 4.x



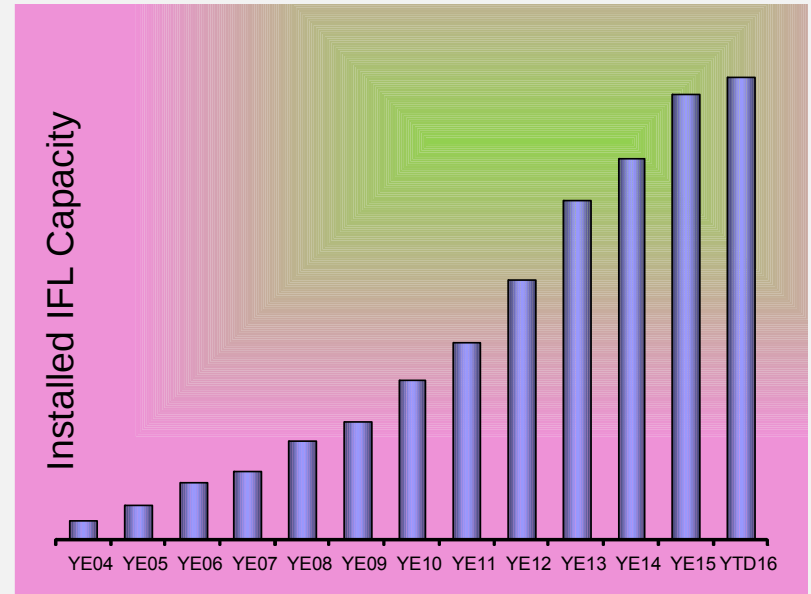
Linux on IBM z Systems in 2Q2016

*Installed Linux MIPS at 42% CAGR**

- 48% of z Systems Customers have IFL's installed as of 2Q16
- 93 of the top 100 z Systems Customers are running Linux on z as of 2Q16 **
- 27.7% of Total installed MIPS run Linux as of 2Q16
- Installed IFL MIPS increased by 10% YTY from 2Q15 to 2Q16
- 36% of all z Systems servers have IFLs
- 62% of new FIE/FIC z Systems accounts run Linux

* Based on YE 2003 to YE 2015 **Top 100 is based on total installed MIPS

Installed Capacity Over Time



Linux on IBM z Systems distributions

What is available today

- **SUSE Linux Enterprise Server 10**
 - GA 17 Jul. 2006; Kernel 2.6.16; GCC 4.1.0
 - SLES 10 SP4: GA 12 Apr. 2011; **EOS 31 Jul. 2013; LTSS: 30 Jul. 2016**
- **SUSE Linux Enterprise Server 11**
 - GA 24 March 2009; Kernel 2.6.27 (SP4: 3.0); GCC 4.3.3 (SP4 4.3.4)
 - SLES 11 SP4: GA 15 Jul. 2015; EOS 31 Mar. 2019; LTSS: 31 Mar. 2022
- **SUSE Linux Enterprise Server 12**
 - GA 27.10.2014; Kernel 3.12; GCC 4.8
 - SLES 12 SP1: GA 15 Dec. 2015;
 - Last SP: EOS 31 Oct. 2024; LTSS: 31 Oct. 2027
- <https://www.suse.com/support/policy.html>
- <https://www.suse.com/lifecycle/>

Red Hat Enterprise Linux

- **Red Hat Enterprise Linux AS 4**
 - GA 14 Feb. 2005; Kernel 2.6.9; GCC 3.4
 - RHEL 4.9: GA 16 Feb. 2011; **EOS 29 Feb. 2012; ELS: 31 Mar. 2017**
- **Red Hat Enterprise Linux AS 5**
 - GA 15 Mar. 2007; Kernel 2.6.18; GCC 4.1
 - RHEL 5.11: GA 16 Sep. 2014; EOS 31 Mar. 2017; ELS: 30 Nov. 2020
- **Red Hat Enterprise Linux AS 6**
 - GA 9.11.2010; Kernel 2.6.32; GCC 4.4
 - RHEL 6.8: GA 10 May 2016
 - Last Update: EOS 30 Nov. 2020; ELS: tbd
- **Red Hat Enterprise Linux AS 7**
 - GA 9.6.2014; Kernel 3.10; GCC 4.8
 - RHEL 7.2: GA 19 Nov. 2015
 - Last Update: EOS 30 Jun. 2024; ELS: tbd
- <https://access.redhat.com/support/policy/updates/errata/>

Ubuntu Server

- Canonical and IBM announced on LinuxCon 2015 (17 Aug. 2015) their plans to create an **Ubuntu** based distribution for **IBM z Systems and LinuxONE**.
 - <http://www-03.ibm.com/press/us/en/pressrelease/47474.wss>
- **Ubuntu Server 16.04 LTS**
 - GA 21 Apr. 2016; EOS: Apr. 2021
 - Kernel 4.4; GCC 5.3
- Ubuntu Lifecycle:
 - Normal 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>
 - https://wiki.ubuntu.com/Kernel/LTSEnablementStack?_ga=1.219828057.1549132454.1460845469

Supported Linux Distributions

Distribution	LinuxONE Emperor	LinuxONE Rockhopper			
	z13	z13s	zEnterprise - zBC12 and zEC12	zEnterprise - z114 and z196	System z10 and System z9
RHEL 7	✓ (1)	✓ (1)	✓ (3)	✓ (3)	✗
RHEL 6	✓ (1)	✓ (1)	✓ (4)	✓	✓
RHEL 5	✓ (1)	✗ (10)	✓ (5)	✓	✓
RHEL 4 (*)	✗	✗	✗	✓ (8)	✓
SLES 12	✓ (2)	✓ (2)	✓	✓	✗
SLES 11	✓ (2)	✓ (2)	✓ (6)	✓	✓
SLES 10 (*)	✗	✗	✓ (7)	✓	✓
SLES 9 (*)	✗	✗	✗	✓ (9)	✓
Ubuntu 16.04	✓	✓	✓	✗	✗



Please check the link for minimum required kernel levels.



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.



Current Linux on IBM z Systems Technology

Key features & functionality already
contained in the SUSE & Red Hat Distributions

Tag legend


- **Supported distributions**

 for SUSE SLES <X> Service Pack <Y>, e.g.  for SLES12 SP1

 for RHEL <x> Update <y>, e.g.  for RHEL7.2

 for Ubuntu x.y, e.g.  for Ubuntu 16.04 LTS

- **Supported environments**

 usable for systems running under LPAR

 usable for guests running under z/VM

 usable for guests running under KVM

- **Vector extension facility (kernel 3.18)**
LPAR
z/VM
KVM
12.1
7.2
16.04
 - Also known as single-instruction, multiple data (**SIMD**)
 - 32 128-bit vector registers are added to the CPU
 - 139 new instructions to operate on the vector registers
 - User space programs can use vectors to speed up all kinds of functions, e.g. string functions, crc checksums, ...

- **CPU multi threading support (> kernel 3.19)**
LPAR
12.1
7.2
16.04
 - Also known as simultaneous multi-threading (**SMT**)
 - Once enabled the multi threading facility provides multiple CPUs for a single core.
 - The CPUs of a core share certain hardware resource such as execution units or caches
 - Avoid idle hardware resources, e.g. while waiting for memory



IBM z13 Support

- **Extended number of AP domains (kernel 3.18)**



- AP crypto domains in the range 0-255 will be detected

- **Crypto Express 5S cards (kernel 4.0)**



- New generation of crypto adapters with improved performance

- **z13 cache aliasing (kernel 4.0)**



- Shared objects mapped to user space need to be aligned to 512KB for optimum performance on z13

IBM z13 – Vector Extension alias SIMD

- **32 vector registers with 128 bits each**

- Register can be split into 16 bytes, 8 shorts, 4 integers, or 2 long integers
- Up to 4 concurrent 32x32 multiply / adds
- Many new vector instructions with many specialized use cases
 - e.g. Vector Galois Field Multiply Sum and Accumulate (VGFMA)

- **Vector registers and floating pointer register partially overlap**

- The program can use either the FPR or the VR with 0-15 at the same time

0	%f0 or %v0 bits 0:63	%v0 bits 64:127
...
15	%f0 or %v0 bits 0:63	%v0 bits 64:127
16	%v0 bits 0:127	
...	...	
31	%v0 bits 0:127	



IBM z13 – Vector Extension alias SIMD

- **Vector instruction example: `size_t strlen(const char *s)`**

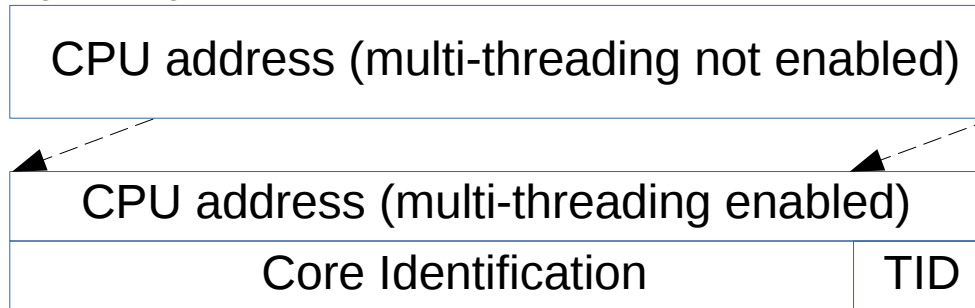
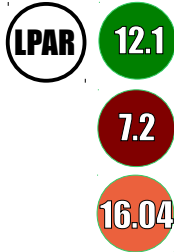
# R2 address of the string, R1 will contain length	
XGR R1,R1	Zero out running index
LOOP: VLBBV16,0(R1,R2),6	Load up to 16 bytes
LCBBR3,0(R1,R2),6	Find how many bytes were loaded
ALGRK R1,R1,R3	Increment length by bytes loaded
VFENEBZ V17,V16,V16	Look for 0 byte
VLVGB R4,V17,7(R0)	Extract index to gpr (16-no match)
CLGR R3, R4	If GLEN <= GPOS have more to search
BRNH LOOP	
SLGRK R1,R1,R3	Subtract off amount loaded
ALGRK R1,R1,R4	Add amount to the zero that was found

- 7 instructions to scan 16 bytes in the body of the `strlen` loop
- Standard implementation uses loop unrolling for 8 bytes
 - 1x 'CLI' + 1x 'BRC' for each byte, 1x 'LA' + 1x 'BRCT' for 8 bytes
 - 34 instructions to scan 16 bytes in the body of the `strlen` loop



IBM z13 – CPU multi threading alias SMT

- **Up to two hardware threads on a single core**
- Known as simultaneous multi-threading (SMT) on z Systems and POWER, or Hyper-threading on x86
- **The operating system needs to opt-in to enable SMT**
- The LPAR code starts a partition with one logical CPU per core
- After the SIGP to enable MT additional logical CPUs are surfaced and the CPU addressing changes

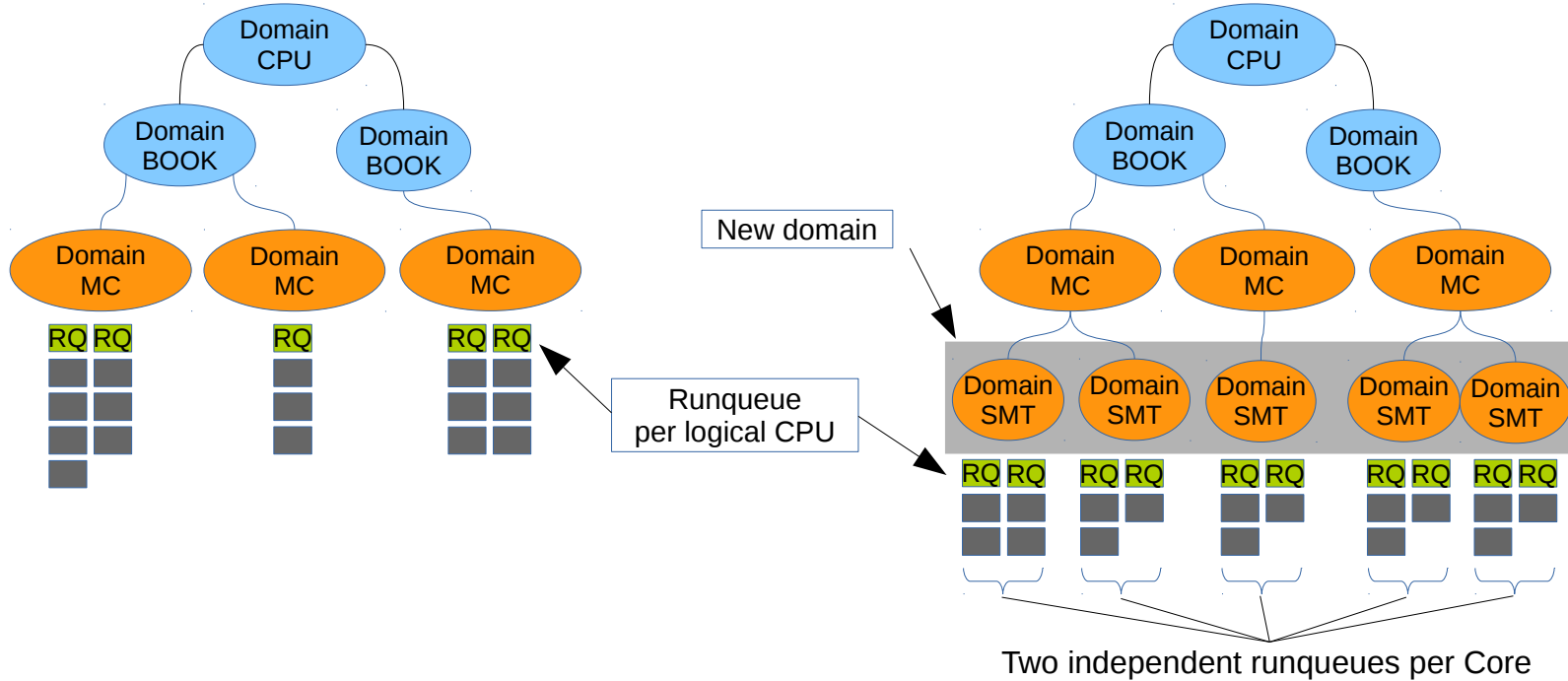


- The additional CPUs work just like “normal” CPUs, but with different performance characteristics

IBM z13 – CPU multi threading alias SMT

- Scheduler CPU domains non-MT vs. MT

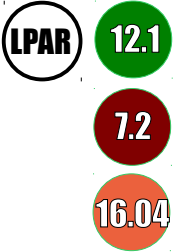
LPAR 12.1
7.2
16.04



IBM z13 – CPU multi threading alias SMT

• CPU time accounting with MT enabled

- New CPU-MF counters: number of cycles with n threads active
- Average thread density is calculated once per tick
- The raw CPU timer deltas are scaled with the average thread density
- The scaled CPU time is available via the 'taskstats' interface
- The standard CPU time values in /proc use the unscaled CPU time

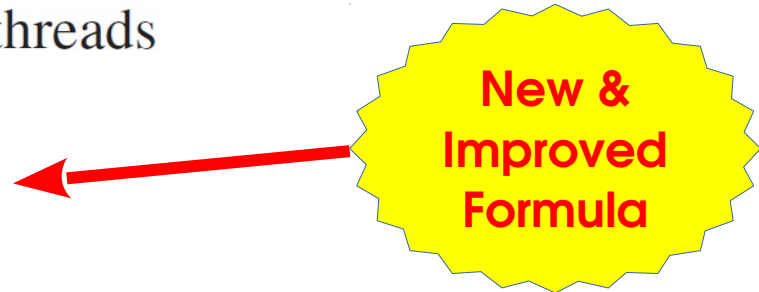


sT_n scaled CPU time

T_n unscaled CPU time

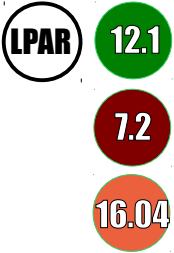
C_i cycle count with i active threads

$$sT_n = T_n * \frac{\sum_{i=1}^n \frac{C_i}{i}}{\sum_{i=1}^n C_i}$$



IBM z13 – CPU multi threading alias SMT

- **Workloads may or may not benefit from MT**
 - Certain processor resource need to be underutilized for MT to be effective
- **Workloads characteristics with a positive effect on MT**
 - A large number of cache misses
 - Long chains of instruction dependencies
 - A large number of branch mis-predictions
- **Workload characteristics with a detrimental effect on MT**
 - A homogeneous instruction mix targeting a scarce function unit
 - Intensive use of the memory system with poor cache locality
 - Applications with poor scaling, in general MT requires more logical CPUs



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

- 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



- **zEC12/zBC12 exploitation CPU (gcc 4.8)**

- Use option `-march=zEC12` to utilize the instructions added with zEC12
- Use option `-mtune=zEC12` to schedule the instructions appropriate for the pipeline of zEC12
- Transactional memory support, Improved branch instructions



- **z13/z13s exploitation CPU (gcc 5.2)**

- Use option `-march=z13` to utilize the instructions added with z13
- Use option `-mtune=z13` to schedule the instructions appropriate for the pipeline of z13
- SLES12SP1 support with the gcc 5.3.1 toolchain module



- **Native PCIe feature cards introduced on zEC12 and zBC12**
 - Plugged into an PCIe I/O drawer
 - Managed by an internal firmware processor (IFP)
 - Device driver for PCIe function is located in the operating system
- **10 GbE RoCE Express, networking card**
 - Uses Infiniband RDMA over Converged Ethernet (RoCE) specification
- **zEDC Express, data compression / decompression**
 - Implements compression as defined by RFC 1951 (DEFLATE)
 - Comparable to “gzip -1”



New kernel features - eBPF

- **eBPF kernel JIT backend (kernel 4.1)**







- The BPF interpreter was originally developed as a network packet filter
- The 32-bit BPF just-in-time compiler improves the speed of the BPF program
- The extended BPF (eBPF) is 64-bit enabled and is intended for more general use, e.g. traffic classification, system call filtering, performance event filtering, and so on.

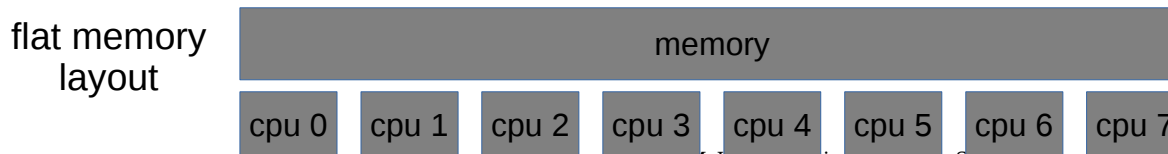
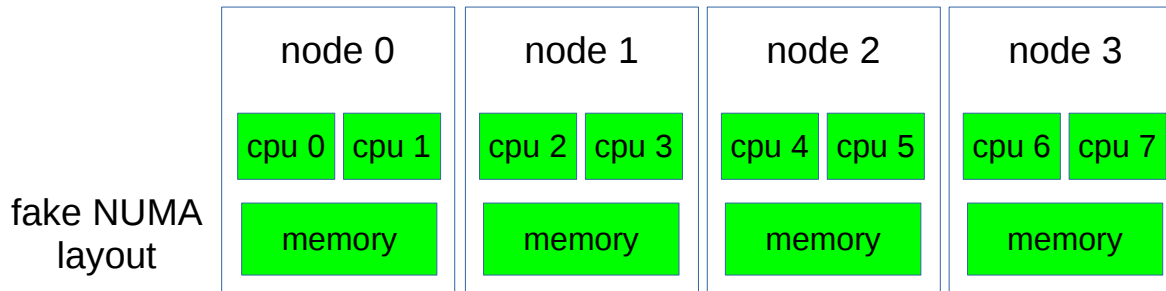
- **Example: tcpdump -i eth0 arp**

```
Initial BPF program
(000) ldh   [12]
(001) jeq   #0x806   jt 2jf 3
(002) ret   #65535
(003) ret   #0
```

```
JITed program in the kernel
stmg      %r6,%r7,72(%r15)      ...
stmg      %r12,%r15,120(%r15)   basr      %r6,%r1
lgr       %r1,%r15              brc       7,3ff818e6122
la        %r13,72(%r15)         lgfi      %r1,2054
aghi      %r15,-616             clgrij    %r14,%r1,7,3ff818e6118
stg       %r1,152(%r15)         llilf     %r14,65535
llgf      %r1,128(%r2)          brc       15,3ff818e6126
s         %r1,132(%r2)          llilf     %r14,0
stg       %r1,168(%r15)         brc       15,3ff818e6126
lg        %r12,216(%r2)         lghi      %r14,0
lgr       %r7,%r2               lgr       %r2,%r14
llilf     %r1,1233228           lmg       %r6,%r7,688(%r15)
lgfi      %r3,12                lmg       %r12,%r15,736(%r15)
...                               bcr       15,%r14
```


New kernel features – NUMA emulation for KVM

- **NUMA emulation splits the available system memory into equal chunks to the selected number of fake NUMA nodes (kernel 4.3)**    
- Initially create as a debug tool to verify NUMA code on non-NUMA systems
- For large systems splitting the memory into multiple nodes has performance advantages
 - Multiple sets of LRU lists avoids contention in the memory management
 - Multiple kswapd processes improve scalability
 - Each node has its own free memory watermarks


















Container Support for Docker

- **Docker provides lightweight containers**
 - Self contained set of files to package an application with all of its dependencies
- **Applications in containers share the OS kernel**
 - No virtualization – no virtualization overhead
- **“Build, Ship, and Run Any App, Anywhere”**
 - One implementation of a container solution
 - Maintained by Docker, Inc.
 - Docker Hub cloud-based registry service, see <https://hub.docker.com>
- **Power tool to build, modify, deploy, run, manage containers**
 - E.g. “docker run hello-world”



Miscellaneous new kernel features

- Hot-patch support for function tracing (kernel 4.0)**





 - Use gcc's hotpatch support to generate better code for ftrace function tracing
 - Each function starts with a six byte nop instruction which will be patched at run-time
- IUCV tty simple wildcard match for the z/VM user IDs (kernel 4.0)**


 - Introduce a wildcard character to filter a range of z/VM user IDs
 - Reduces the number of filter entries of the IUCV tty setup
- Add z/VM CPU performance metrics interface (kernel 4.0)**


 - Allows to read the CPU performance data provided by z/VM via diagnose 0x0C
 - A new binary debugfs file `/sys/kernel/debug/s390_hypfs/diag_0c` is added for user space access with `open/read/close`
- In-kernel crypto: DRBG support (kernel 4.1)**






 - Deterministic random bit generator alias RNG, PRNG

Miscellaneous new kernel features

- **qeth: enable default drain (kernel 4.2)** LPAR z/VM 16.04

 - Extend bridge port functionality to OSA devices
 - Requires OSA firmware with support for bridge port
- **Enable magic sys requests in SCLP VT220 console (kernel 4.3)** LPAR z/VM KVM 16.04

 - “Ctrl+o” followed by the character for the the sys-request will trigger the magic
- **Implement proper Link Incident Record handling (kernel 4.3)** LPAR z/VM 16.04

 - Helps the operator to identify degraded or non-operational FICON connections
- **Support for IPL Device in Any Sub-Channel Set (kernel 4.4)** LPAR z/VM 16.04

 - Allows to boot the OS from a device with an address '0.x.yyyy' with x != 0
- **Add a statistic for diagnose calls (kernel 4.4)** LPAR z/VM KVM 16.04

 - Provide the number of diagnose calls per CPU via '/sys/kernel/debug/diag_stat'

Miscellaneous new kernel features – diag statistics

- Useful to find congestion problems
 - Watch the values for diag 044 and diag 09c
 - The high value on CPU #0 is due to a timing loop at IPL



```
# cat /sys/kernel/debug/diag_stat
      CPU0      CPU1      CPU2      CPU3
diag 008:         0         0         0         0  Console Function
diag 00c:         0         0         0         0  Pseudo Timer
diag 010:         0         0         0         0  Release Pages
diag 014:         0         0         0         0  Spool File Services
diag 044:    663700         1         1         1  Voluntary Timeslice End
diag 064:         0         0         0         0  NSS Manipulation
diag 09c:         3         2         3         1  Relinquish Timeslice
diag 0dc:         0         0         0         0  Appldata Control
diag 204:         0         0         0         0  Logical-CPU Utilization
diag 210:         0         0         0         0  Device Information
diag 224:         0         0         0         0  EBCDIC-Name Table
diag 250:         0         0         0         0  Block I/O
diag 258:         1         0         0         0  Page-Reference Services
diag 288:         0         0         0         0  Time Bomb
diag 2c4:         0         0         0         0  FTP Services
...

```

New kernel features – networking improvements

- **Enable Layer 2 offloads (kernel 4.4)**
 - Enable large send/receive and checksum offload of OSA in layer 2 mode
- **IOMMU API Support (kernel 4.4)**
 - Allows to map physical memory to arbitrary virtual DMA addresses
 - Required for e.g. PCI pass-through via the vfio framework



Current Linux on IBM z Systems Technology

Platform features & functionality from IBM

IBM XL C/C++ for Linux on z Systems, V1.2

High-performance compiler now delivered



- **XL C/C++ for Linux on z Systems offers:**
 - **Increased return on hardware investments** for improved application performance with leading-edge optimization technology and exploitation of the latest z Systems hardware.
 - **The Clang infrastructure in the front end with advanced optimization technology** in the IBM compiler back end.
 - **Highly tuned math libraries**, including Mathematical Acceleration Subsystem (MASS), Basic Linear Algebraic Subprograms (BLAS), and Automatically Tuned Linear Algebra Software (ATLAS).
 - **Standards compliance** with support for international C and C++ programming language standards and GNU C/C++ compatibility extensions for ease of application migration to IBM z Systems.
 - **Software requirements**
 - Red Hat Enterprise Linux for IBM System z 7
 - Red Hat Enterprise Linux for IBM System z 6.3
 - SUSE Linux Enterprise Server for System z 12
 - SUSE Linux Enterprise Server for System z 11 SP3
 - Ubuntu 16.04 LTS

Available since Feb 2015

XL C/C++ for Linux on z Systems
60 day no-charge trial

Download software



More information:

• [Data sheet:](#)

• [Café Community & Forum](#)

IBM XL C/C++ for Linux on z Systems, V1.2



High-performance XL C/C++ compiler for Linux on z Systems

IBM® XL C/C++ for Linux on z Systems® V1.2 is a C/C++ compiler for application development that takes advantage of the latest IBM z Systems servers that run on selected Linux distributions. This product strengthens the platform, exploits the z Systems environment, and provides superior performance by generating highly optimized code for execution on IBM z Systems. It is built on the performance gains from many years of IBM compiler optimization experience with existing XL C/C++ compilers that are available for IBM z/OS®, IBM z/VM®, IBM AIX®, and Linux on IBM Power Systems®. With XL C/C++ for Linux on z Systems, you can create and port applications for execution on the next generation of IBM z Systems supporting selected Linux distributions while maximizing hardware utilization with fast performance.

Highlights

- XL C/C++ for Linux on z Systems, V1.2 features the following:
- Vector single instruction, multiple data (SIMD) and auto-SIMD support for the new vector extension facility (SMD) instructions available on IBM AIX® servers
 - Generation of highly optimized code exploiting z Systems servers
 - Support of programming language standards, including C11 compliance and additional features of the latest C++11 and C++14 standards
 - High level of source compatibility with GNU Compiler Collection (GCC) while providing binary coexistence
 - Enhancements to the ATLAS and IBM MASS libraries to take advantage of the vector facility in z13 for potential performance improvements

Vector SIMD and auto-SIMD support

XL C/C++ for Linux on z Systems, V1.2 supports the vector single instruction, multiple data (SMD) and auto-SIMD instructions that are provided with the z13 vector extension facility and corresponding vector programming language extensions. This exploitation of the vector facility in z13 helps improve the performance of your XL C/C++ applications.

Generation of highly optimized code

As the newest addition to the IBM compiler family, IBM XL C/C++ for Linux on z Systems, V1.2 brings mature IBM compiler technology to Linux distributions running on IBM z Systems servers. XL C/C++ supports generation of highly optimized code exploiting z Systems servers, adheres to programming standards, and provides GNU Compiler Collection (GCC) compatibility that allows you to easily port your applications to Linux distributions running on z Systems servers.

The Clang Infrastructure

XL C/C++ for Linux on z Systems leverages the Clang infrastructure from the open source community for a portion of its compiler front end. Clang is a component of the LLVM open source compiler and toolchain project, and provides the C and C++ language family front end for LLVM. XL C/C++ for Linux on z Systems combines the Clang front end infrastructure with the advanced optimization technology in the IBM compiler back end.

New architecture and tune compiler options for the z Systems technology

XL C/C++ for Linux on z Systems supports the generation of z Systems hardware running SUSE Linux Enterprise Server 11 (SLES 11), SLES 12, Red Hat Enterprise Linux 6 (RHEL 6), and RHEL 7.

ibm.com/software/products/en/czlinux

KVM for IBM z Systems

- First release 1.1.0 in 9/2015, update to 1.1.1 in 3/2016
 - Based on kernel 3.10 and qemu 2.3
 - Kernel fixes up to version 4.1, and a lot of fixes from qemu 2.4
 - Run Linux guests on a Linux host
 - Requires a zEC12 or newer machine
 - Supported distributions
 - SUSE Linux Enterprise Server (SLES) 12 SP1
 - Ubuntu 16.04 LTS
- Links
 - <http://www.ibm.com/systems/z/solutions/virtualization/kvm/>
 - <http://www.redbooks.ibm.com/abstracts/sg248332.html?Open>

IT Infrastructure > z Systems (Mainframes) > Solutions > Virtualization >

KVM for IBM z Systems

Open source virtualization hypervisor

KVM for IBM z Systems provides open source virtualization for IBM z Systems and the LinuxONE platforms. Using the combination of KVM virtualization and IBM z Systems and LinuxONE, you have the performance and flexibility to address the requirements of multiple, differing Linux workloads. KVM's open source virtualization on IBM z Systems and LinuxONE allows businesses to reduce costs by deploying fewer systems to run more workloads, sharing resources and improving service levels to meet demand.

Highlights

- **Open virtualization:** Take advantage of the performance, scalability and security built into Linux and KVM and gain a cost effective alternative to proprietary x86 virtualization.
- **Quality of service:** Gain easy provisioning for predictability of delivery of service at high utilization rate.
- **Operational efficiencies:** Use familiar Linux interface to gain greater operational efficiency.

Learn more

- [Announcement letter](#)
- [Data sheet \(192KB\)](#)
- [FAQ \(1.6MB\)](#)
- [Technical Information \(616KB\)](#)
- [Redbook](#)

Blockchain on Linux on IBM z Systems

- **Blockchain is a technology for a new generation of transactional applications that establishes trust, accountability and transparency while streamlining business processes.**
- **2 of the 3 IBM Blockchain offerings use Linux on IBM z Systems:**
 - The **High Security plan of IBM Blockchain on Bluemix – cloud**
 - **IBM Blockchain on IBM z Systems – on premise**
- Both profiting from **qualities of z Systems** (security, availability, performance):
 - Co-location with business data
 - Isolated partitions in memory keep ledgers separate and secure
 - High availability and scalability of IBM z Systems
 - Hardware encryption with built-in accelerators
 - Reduced data center footprint, simplified management, energy savings

IBM Spectrum Scale 4.2.1

- **IBM Spectrum Scale is software-defined storage built on IBM General Parallel Filesystem (GPFS) for high performance, large scale workloads on-premises or in the cloud.**
- **Features of Advance Edition:**
 - Asynchronous multisite disaster recovery (DR), enabling active/passive configuration at the fileset level
 - Information lifecycle management (ILM)
 - Support of the IBM Spectrum Protect™ v7.1.4 backup- archive and Space Management client
 - Support for ECKD DASD and FCP attached SCSI disks
 - Heterogeneous clusters with client nodes without local storage access running Linux distributions from Red Hat and SUSE on x86 and Power®, and AIX® on Power
 - Support of the storage systems: IBM System Storage® DS8000® series, IBM Storwize® V7000 Disk Systems, IBM XIV® Storage Systems, IBM FlashSystem™ systems, IBM System Storage SAN Volume Controller (SVC), all other storage hardware using ECKD, and any storage system connected via SVC
- **Supported distributions:**
 - Red Hat Enterprise Linux 6.5 or newer
 - SUSE Linux Enterprise Server 11 SP3 or newer
 - Ubuntu 16.04 LTS Server

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

New features – latest kernel versions

- **Query host access to DASD volume (kernel 4.7)**
 - Allow applications to query if a DASD volume is in use by another operating system
- **Quick format mode for use with dasdfmt (kernel 4.7)**
 - Allows quick format mode to dasdfmt, for cases where a DASD device might have been formatted earlier and could potentially be re-initialized very easily (just VTOC and Label)
- **Add drawer scheduling domain (> kernel 4.7)**
 - Reflect the drawer / node structure of z13/z13s to the Linux scheduler
- **Vector optimization for CRC32 (> kernel 4.7)**
 - Speed up the in-kernel CRC32 code by use of vector instructions
- **2GB pages for hugetlbfs (> kernel 4.7)**
 - Extend the huge page support to allow 2GB huge pages next to 1MB large pages
 - Promises to speed up Java with large heap sizes and databases with big SGAs

New kernel (anti) features – kernel removals

- **31-bit kernel support removed with kernel 4.1**
 - 31-bit application will continue to work, only the 31-bit kernel is discontinued
 - A 64-bit kernel can be used in combination with a 100% 31-bit user space
- **The claw network driver removed with kernel 4.1**
 - Common Link Access for Workstation, used some old RS/6000s, Cisco Routers (CIP) and 3172 devices
 - We could not find a single user of this device driver
- **Emulated large pages removed with kernel 4.2**
- **Move OSA portname into deprecated status with kernel 4.4**
- **Remove sclp_cpi module in favor of sysfs interface with kernel 4.5**
- **Time synchronization with ETR will be removed with kernel > 4.7**

Linux on IBM z Systems info at:
ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_main.html

Documentation news – Updates available

- **Linux Distributions**

- SUSE: SLES12 SP1, SLES11 SP 3
- RedHat: RHEL 7.2, RHEL 6.4

- **Upstream Linux 4.0, 4.1, 4.2, 4.3, 4.4**

- ibm.com/developerworks/linux/linux390/documentation_dev.html

- **KVM**

- KVM Virtual Server Quick Start
- KVM Virtual Server Management
- Device Driver, Features and Commands for Linux as a KVM Guest
- Installing SUSE Linux Enterprise Server 12 as a KVM Guest

- **How to Documents**

- How to Set up a Terminal Server Environment

Linux on System z



Device Drivers, Features, and Commands

Development stream (Kernel 37)

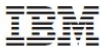
Linux on System z



How to Improve Performance with PAV

Development stream (Kernel 2635)

Linux on System z



Kernel Messages

Development stream (Kernel 37)

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 IBM z Systems distributions**
 - It is **the** essential tool chain for Linux on IBM z Systems
 - It contains everything from the boot loader to dump related tools for a system crash analysis
 - Latest version dated 08/2016 is 1.36.0
- **This software package is contained in all major (and IBM supported) enterprise Linux distributions which support s390**
 - RedHat Enterprise Linux version 5, 6, and 7
 - SuSE Linux Enterprise Server version 10, 11, and 12
- **Website:**
<http://www.ibm.com/developerworks/linux/linux390/s390-tools.html>
- **Feedback: linux390@de.ibm.com**

s390-tools package – the content

BOOT

zipl

DISPLAY

lscss
lschp
lsdasd
lsluns
lsqeth
lsreipl
lsshut
lsscm
lstape
lszcrypt
lszfcf
lshmc
lsmem
lscpumf
lszdev

CHANGE

chccwdev
chchp
chreipl
chshut
chzcrypt
chmem
chcpumf
chzdev

DASD

dasdfmt
dasdinfo
dasdstat
dasdview
fdasd
tunedasd

FILESYSTEM

hmcdrvfs
zdsfs

NETWORK

ip_watcher
osasnmpd
qetharp
qethconf
qethqoat
znetconf

TAPE

tape390_display
tape390_crypt

MONITOR

mon_fsstatd
mon_procd
ziomon
ziorep_config
ziorep_traffic
ziorep_utilization
hyptop
cpacfstats

DUMP & DEBUG

dbginfo
dumpconf
zfcpdump
zfcpdbf
zgetdump
scsi_logging_level

z/VM

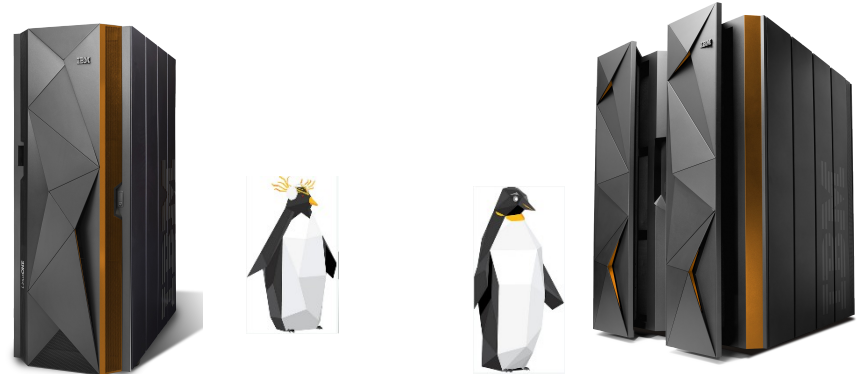
vmconvert
vmcp
vmur
cmsfs-fuse

MISC.

cio_ignore
cpuplugd
chiucvallow
iucvconn
iucvtty
ts-shell
ttyrun

Thank you very much for your attention

Questions?



Dr. Manfred Gnirss

Schönaicher Strasse 220
71032 Böblingen, Germany

IBM Client Center
IBM Germany Lab

Phone +49 (0)7031-16-4093
gnirss@de.ibm.com