



**Linux auf z**  
Dr. Manfred Gnirß / Arwed Tschoeke  
Client Center Böblingen

© 2016 IBM Corporation

## Linux on IBM z Systems



## Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

BladeCenter*	FICON*	OMEGAMON*	RACF*	System z9*	zSecure
DB2*	GDPS*	Performance Toolkit for VM	Storwize*	System z10*	z/VM*
DS6000*	HyperSockets	Power*	System Storage*	Tivoli*	z Systems*
DS8000*	HyperSwap	PowerVM	System x*	zEnterprise*	
ECKD	IBM z13*	PR/SM	System z*	zOS*	

\* Registered trademarks of IBM Corporation

### The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Call Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, 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.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

OpenStack is a trademark of OpenStack LLC. The OpenStack trademark policy is available on the [OpenStack website](http://OpenStack.org/webinars).

TEALEAF is a registered trademark of Tealeaf, an IBM Company.

Windows Server and the Windows logo are trademarks of the Microsoft group of countries.

Worklight is a trademark or registered trademark of Worklight, an IBM Company.

UNIX is a registered trademark of The Open Group in the United States and other countries.

\* Other product and service names might be trademarks of IBM or other companies.

### Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

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 subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIPs, zAAPs, and IFLs) ("SEs"). 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](http://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 SE 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.



## Dank an

- sehr viele Kollegen, v.a. Martin Schwidefsky

## Agenda


- Warm Up
- Distributions
- Features & Functionalities within the Distros
- Upstream Features (not yet within the distros)

Linux on IBM z Systems

SE  
Service Element  
Deutsche Region

IBM

## Warm Up



5

© 2016 IBM Corporation

Linux on IBM z Systems

SE  
Service Element  
Deutsche Region

IBM

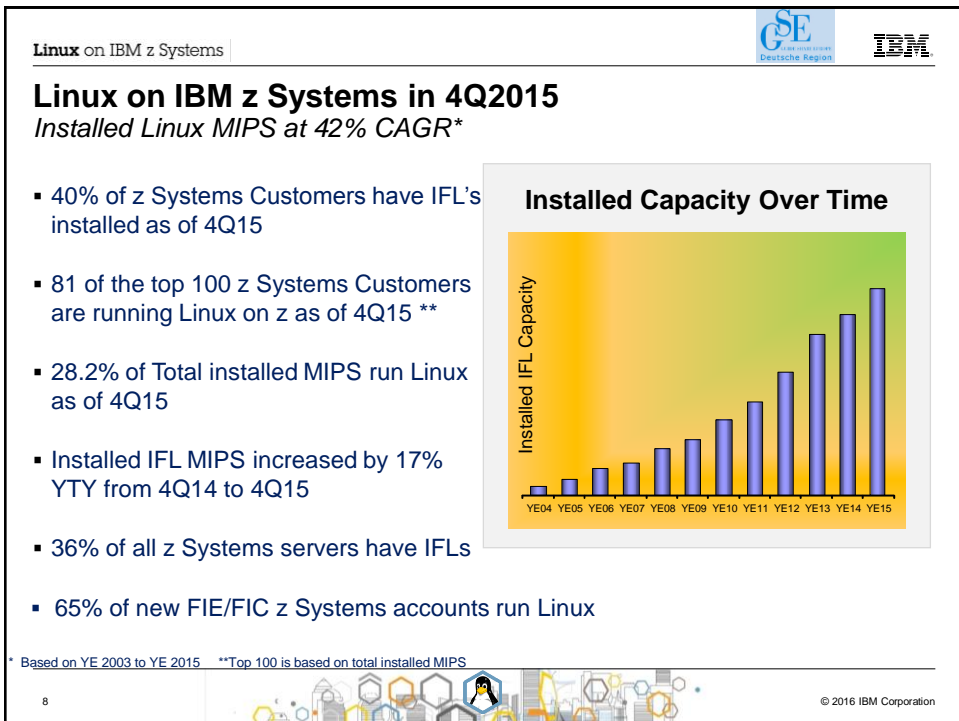
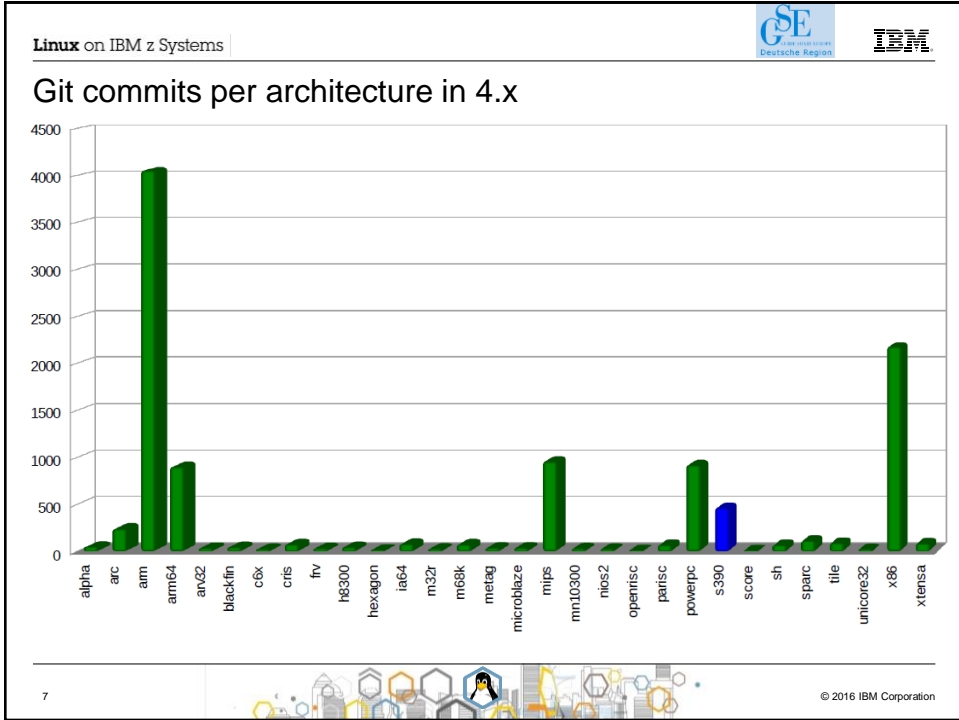
## Fun facts around Linux on IBM z Systems

- Linux kernel v4.5 has 21,153,740 lines of code  
How many lines of code had the initial patch for s390 ?  
36,811 lines of code (patch size)
- How many lines of code are specific to s390 in v4.5 ?  
210,448 lines of code (1.02%)
- How many lines of assembler code are in v4.5 ?  
401,225 lines in 1,419 files, 4,594 lines in 30 files specific to s390 (1.16%)
- How many individual developers contributes to Linux in 2015 ?  
4022 individuals, 1324 with a single commit, 57 contributed directly to s390(81 contributors including wide changes, typo fixes, janitor patches, ...)

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

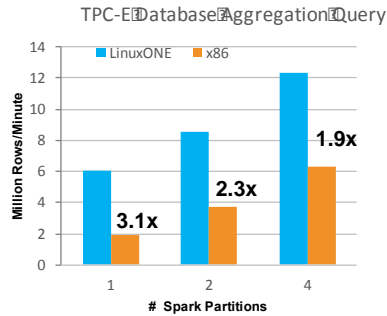
6

© 2016 IBM Corporation



## Analytics workloads run faster on our platform

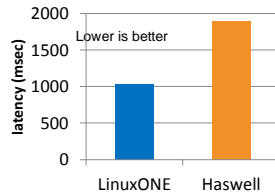
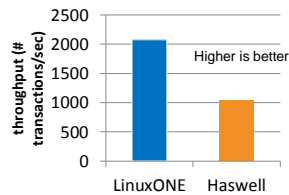
- Test using independent OLTP brokerage database and analytical queries
  - Apache Spark with map-reduce script performs fetch, query and aggregation
    - 1 master and 4 worker JVMs
  - Oracle used for OLTP database, hosted on our platform
    - 348 M rows of brokerage trade data
- Spark co-located on our platform drove **up to 3x** more throughput than Spark running off platform on x86



Test ran on LinuxONE guest, RHEL 6.5 on LinuxONE/VM 6.3 with 32 vCPUs (16 cores with SMT) and 512GB memory, Spark 1.5.0, IBM JDK1.7-SR3  
 Test ran on virtualized x86 with 16 cores on Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.30GHz, 512GB memory running SLES 11.3, Spark 1.4.1, JDK1.8  
 TPC-E 20K scale on Oracle database V12  
 Scale map-reduce script (aggregation query) on TPC-E 20K scale table using 1 master and 4 worker JVMs  
 Out-of-box, default parameters, no tuning

## Extreme Virtualization with Docker! *New Results!!*

- **Containers:** simple way to build and deploy SW with Docker currently leading framework
- LinuxONE runs **4096 active** Docker containers **2.0x** better than Haswell!
  - Workload: 50% WAS Liberty 8.5.5.2, IBM JDK 8.0, Apache Solr 4.10.0, and 50% busybox httpd server
  - With **GOLANG** now avail on z!
- LinuxONE ran **10K** Docker containers with mixed (heavy & light) workloads
  - Unable to get comparable results for more than 5K containers on Haswell
  - Workload: WAS Liberty 8.5.5.2, IBM JDK 8.0, Apache Solr 4.10.0 plus busybox httpd server (no NAT)
- Single LinuxONE Emperor ran more than **1 Million** light Docker containers
  - Workload: busybox httpd server (no NAT)

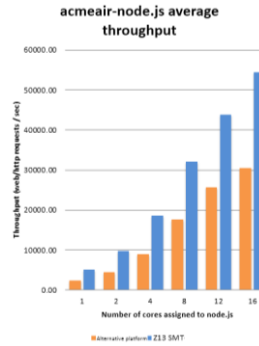


*The throughput and response-time for a single Linux host running 4096 containers*

## Industry leading runtime capabilities with

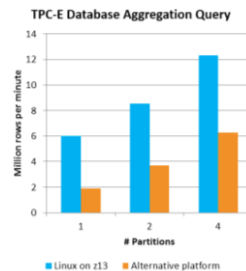
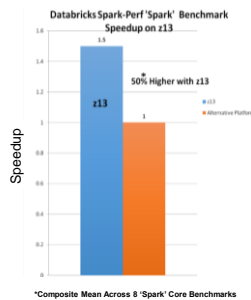


- High Performance JavaScript for LinuxONE and IBM z
  - Up to **2.1x** more RESTful web interactions with AcmeAir in node.js with Apache JMeter benchmark setup



## Spark

- Up to 1.5x faster insights for real-time analytics using Spark's core primitives
- Up to 1.5x more data processed for model building leading to real-time insights with higher accuracy within a given batch window
- Co-locate Spark with non IBM Database on LinuxONE outperforms running Spark off-platform up to 3x for aggregation analytical query
  - e.g. Operational Analytics for a Brokerage running reports on top of OLTP Trading data



Linux on IBM z Systems

Linux your Way: Greater flexibility and choice

**Distributions:** ubuntu, redhat, SUSE, KVM, LPAR

**Languages:** python, Ruby, php, Erlang, Scala, Clojure, JS, Java, OCaml, XMLSec Library, Xerces, doxygen, WordPress

**Runtimes:** node, Rails, ZF, OpenJDK, Apache HTTP Server, Jenkins, fluentd, ANSIBLE, cAdvisor

**Management:** docker, CHEF, openstack, vmware vRealize, Cloud Manager, puppet, SALTSTACK, JMeter, HAProxy, NGINX, Apache ZooKeeper

**Other:** Drupal, kafka, RabbitMQ, Joomla!, Apache Solr, SUGARCRM, Magento, WildFly

**Database:** MariaDB, mongoDB, PostgreSQL, cassandra, CouchDB, MySQL, Oracle, DB2, IBM Cloudant, redis, APACHE GEODE

**Analytics:** Spark, Hadoop, elasticsearch, logstash, Kibana, IBM InfoSphere BigInsights, SPSS, COGNOS

13 © 2016 IBM Corporation

Linux on IBM z Systems

Open Source Ecosystem

- Recipes for building the software on Linux
- Pointers to binaries if available
- Source code repositories and build instructions maintained on GitHub: <https://github.com/linux-on-ibm-z/docs/wiki/>

Open to every one interested in Linux

Linux on Z open source list

Package	SLES 12.x	RHEL 7.x	SLES 11.x	RHEL 6.x	Comments
MariaDB	NA	NA	10.0	10.0	
Maven	3.2.5	3.2.5	NA	NA	
MongoDB	2.4.92.6.6	2.4.92.6.6	3.0	3.0	
MySQL	5.6.24	5.6.24	NA	NA	
Node.js	1.21.1	1.21.1	1.21.1	1.21.1	
OCaml	4.02.1	4.02.1	4.02.1	4.02.1	
PostgreSQL	9.4	9.4	9.4	9.4	
Protobuf	2.6.1	2.6.1	NA	NA	
Puppet	4.1.0	4.1.0	4.1.0	4.1.0	
Python	2.7.9, 3.4.3	2.7.9, 3.4.3	2.7.9, 3.4.3	2.7.9, 3.4.3	
RabbitMQ	3.5.0	3.5.0	3.5.0	3.5.0	
Rails	4.2.1	4.2.1	4.2.1	4.2.1	
Ruby	2.2.1	2.2.1	2.2.1	2.2.1	
Snappy-Java	NA	NA	1.1.2	1.1.2	

**Docker**

Installation


Download for the current Docker package

Date	File	Package	Download Link
2016-03-03	docker	docker-suse12-20160303.tar.gz	File repository for Linux on z/OS
	docker	docker-rhel7-20160303.tar.gz	File repository for Linux on z/OS
2016-03-03	docker	docker-suse11-20160303.tar.gz	File repository for Linux on z/OS
	docker	docker-rhel6-20160303.tar.gz	File repository for Linux on z/OS


[ibm.com/developerworks/community/groups/community/lozopensource](https://ibm.com/developerworks/community/groups/community/lozopensource)

14 © 2016 IBM Corporation

Linux on IBM z Systems




# Distributions



15

© 2016 IBM Corporation

Linux on IBM z Systems



## Linux on IBM z Systems distributions - SUSE

- **SUSE Linux Enterprise Server 10**
  - 07/2006 SLES10 GA: Kernel 2.6.16, GCC 4.1.0
  - 04/2011 SLES10 SP4
- **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
- **SUSE Linux Enterprise Server 12**
  - 10/2014 SLES12 GA: Kernel 3.12, GCC 4.8
  - 12/2015 SLES12 SP1

16

© 2016 IBM Corporation



## Linux on IBM z Systems distributions – Red Hat

### ▪ Red Hat Enterprise Linux AS 5

- 03/2007 RHEL5 GA: Kernel 2.6.18, GCC 4.1.0
- 09/2014 RHEL5 Update 11

### ▪ Red Hat Enterprise Linux AS 6

- 11/2010 RHEL6 GA: Kernel 2.6.32, GCC 4.4.0
- 07/2015 RHEL6 Update 7

### ▪ Red Hat Enterprise Linux AS 7

- 06/2014 RHEL7 GA: Kernel 3.10, GCC 4.8
- 11/2015 RHEL7 Update 2

## Linux on IBM z Systems distributions

### ▪ Ubuntu

- Canonical and IBM announced plans to create an Ubuntu based distribution on LinuxCon 2015 in Seattle

### ▪ Others

- Debian, Slackware,
- Support may be available by some third party

## Supported Linux Distributions <http://www-03.ibm.com/systems/z/os/linux/resources/testedplatforms.html>

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

✓ 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. Please check the [IBM exception letter](#) for important information regarding your server. Updates or service packs applied to the distribution are also supported. Please check with your service provider which kernel-levels are currently in support.

## Features & Functionalities within the Distros



## IBM z13/z13s Support



- **Vector extension facility (kernel 3.18)**
  - 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)**
  - 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/z13s 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/z13s – 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/z13s – Vector Extension alias SIMD



- **Vector instruction example: size\_t strlen(const char \*s)**

```
# R2 address of the string, R1 will contain length
LOOP:  VLBB  XGR   R1,R1           Zero out running index
        VL16,0(R1,R2),6      Load up to 16 bytes
        LCBB  R3,0(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
```

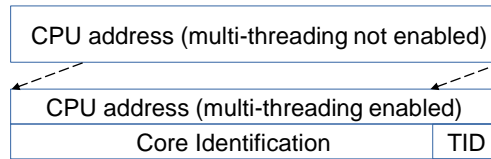
7 instructions to scan 16 bytes in the body of the strlen loop

- 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/z13s – 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/z13s – CPU multi threading alias SMT



- **Example lscpu output with MT enabled**

```
# lscpu -e
CPU BOOK SOCKET CORE L1d:L1i:L2d:L2i ONLINE CONFIGURED POLARIZATION ADDRESS
0 0 0 0 0:0:0:0 yes yes horizontal 0
1 0 0 0 1:1:1:1 yes yes horizontal 1
2 0 1 1 2:2:2:2 yes yes horizontal 2
3 0 1 1 3:3:3:3 yes yes horizontal 3
4 0 1 2 4:4:4:4 yes yes horizontal 4
5 0 1 2 5:5:5:5 yes yes horizontal 5
6 0 1 3 6:6:6:6 yes yes horizontal 6
7 0 1 3 7:7:7:7 yes yes horizontal 7
8 0 1 4 8:8:8:8 yes yes horizontal 8
9 0 1 4 9:9:9:9 yes yes horizontal 9
```

# IBM z13/z13s – CPU multi threading alias SMT

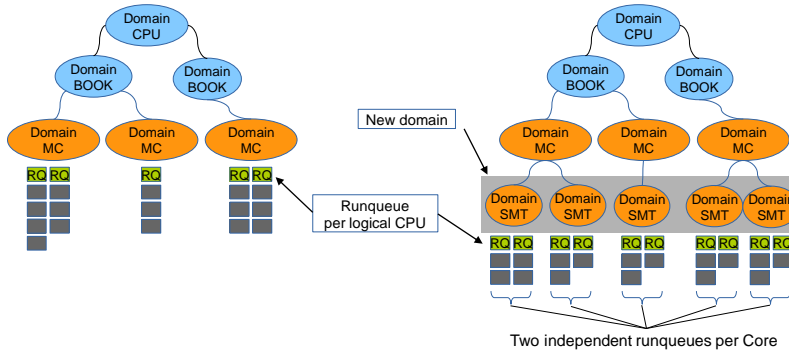


7.2



12.1

## • Scheduler CPU domains non-MT vs. MT



# IBM z13/z13s – CPU multi threading alias SMT



7.2



12.1

## • CPU time accounting with MT enabled

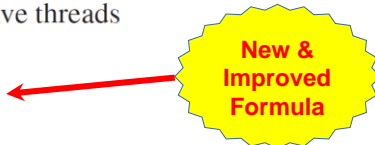
- New CPU-MF counters: number of cycles with n threads active
- Average thread density is calculated once per tick
- The raw CPU timer deltas are scaled with the average thread density
- The scaled CPU time is available via the 'taskstats' interface
- The standard CPU time values in /proc use the unscaled 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/z13s – 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

## Compiler Toolchain

### ▪ 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

## PCI Support

- **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”



## 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”



## Misc Features

- **SE/HMC filesystem (kernel 3.18)**

- Mount the HMC media drive as a read-only Linux file system
- Main use case is the installation of a distribution from the HMC DVD drive



- **Auto port scan resiliency for zfcpl (kernel 3.19)**

- Improves the Fibre Channel port scan behaviour



- **In-kernel crypto: DRBG support (kernel 4.1)**

- Deterministic random bit generator alias RNG, PRNG



- **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



## Upstream Features (not yet within the distros)



## 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**

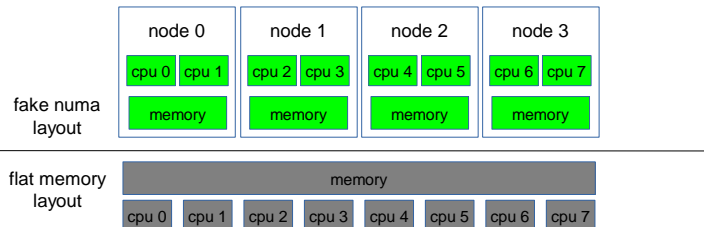
Initial BPF program	JITed program in the kernel
(000) ldh [12]	stmg %r6,%r7,72(%r15) ...
(001) jeq #0x806 jt 2jf 3	stmg %r12,%r15,120(%r15) basr %r6,%r1
(002) ret #65535	lgr %r1,%r15 brc 7,3ff818e6122
(003) ret #0	la %r13,72(%r15) lgfi %r1,2054
	aghi %r15,-616 clgrij %r14,%r1,7,3ff818e6118
	stg %r1,152(%r15) 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

35

© 2016 IBM Corporation

## 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 kswpd processes improve scalability
    - Each node has its own free memory watermarks



36

© 2016 IBM Corporation

## New kernel features – core kernel

- **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`
- **Enable magic sys requests in SCLP VT220 console (kernel 4.3)**
  - “Ctrl+o” followed by the character for the the sys-request will trigger the magic
- **Add a statistic for diagnose calls (kernel 4.4)**
  - Provide the number of diagnose calls per CPU via `/sys/kernel/debug/diag_stat'`

## New kernel features – I/O improvements

- **Handle multiple SCM requests in one HW requests (kernel 3.19)**
  - Do up to 8 block layer requests per HW request to improve performance
- **PCI memory access system calls (kernel 3.19)**
  - Emulate PCI memory access from user space by using system calls
- **Implement proper Link Incident Record handling (kernel 4.3)**
  - Helps the operator to identify degraded or non-operational FICON connections
- **Support for IPL Device in Any Sub-Channel Set (kernel 4.4)**
  - Allows to boot the OS from a device with an address `'0.x.yyyy'` with `x != 0`
- **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

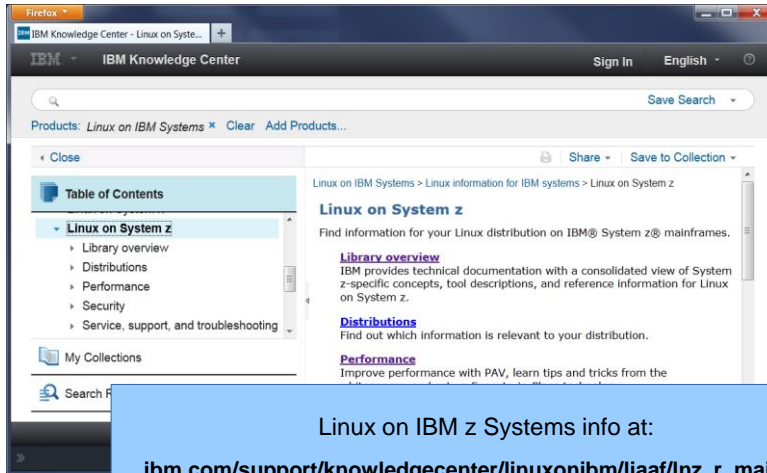
## New kernel features – networking improvements

- **qeth: enable default drain (kernel 4.2)**
  - Extend bridge port functionality to OSA devices
  - Requires OSA firmware with support for bridge port
- **Enable Layer 2 offloads (kernel 4.4)**
  - Enable large send/receive and checksum offload of OSA in layer 2 mode

## 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**
  - The option has no effect, remove dead parameter

## Knowledge center




Linux on IBM z Systems info at:

[ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz\\_r\\_main.html](http://ibm.com/support/knowledgecenter/linuxonibm/liaaf/lnz_r_main.html)

## Linux on IBM z Systems

### 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**
  - [ibm.com/developerworks/linux/linux390/documentation\\_dev.html](http://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 3.7)

Linux on System z 

How to Improve Performance with PAV

Development stream (Kernel 26.05)

Linux on System z 

Kernel Messages

Development stream (Kernel 3.7)

## 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 04/2015 is 1.28.0
- **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

chccwdev chchp chreipl chshut chcrypt chmem CHANGE	dasdfmt dasdinfo dasdstat dasdview fdasd tunedasd DASD	dbginfo dumpconf zfcpdump zfcpdbf zgetdump scsi_logging_level DUMP & DEBUG
lscss lschp lsdasd lsluns lsqeth lsreipl lsshut lstape lszcrypt lszfcp lsmem DISPLAY	mon_fsstatd mon_procd ziomon hyttop MONITOR	vmconvert vmcp vmur cms-fuse z/VM
hmcdrvfs zdsfs FILESYSTEM	ip_watcher osasnmpd qetharp qethconf qethqcat NETWORK	cpuplugd iucvconn iucvtty ts-shell ttyrun MISC
	tape390_display tape390_crypt TAPE	ziapl BOOT



**Dr. Manfred Gnirß**  
**Arwed Tschoeke**

IBM Client Center –  
Systems and Software –  
z ATS  
IBM Germany Lab

Schoenaicher Str. 220  
D-71032 Boeblingen

Phone +49 (0) 171 863 7780  
gnirss@de.ibm.com  
arwed.tschoeke@de.ibm.com