2021 Linux on IBM Z and LinuxONE Virtual Client Workshop

July 12-16 Americas & EMEA
July 27-29 APAC

Boosting TCP Networking Performance on IBM Z and LinuxONE with SMC-Dv2

Stefan Raspl
Linux on IBM Z Development

Trademarks

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

AIX*	DB2*	HiperSockets*	MQSeries*	PowerHA*	RMF	System z*	zEnterprise*	z/VM*
BladeCenter*	DFSMS	HyperSwap	NetView*	PR/SM	Smarter Planet*	System z10*	z10	z/VSE*
CICS*	EASY Tier	IMS	OMEGAMON*	PureSystems	Storwize*	Tivoli*	z10 EC	
Cognos*	FICON*	InfiniBand*	Parallel Sysplex*	Rational*	System Storage*	WebSphere*	z/OS*	
DataPower*	GDPS*	Lotus*	POWER7*	RACF*	System x*	XIV*		

* 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.
Cell 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, 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.
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.

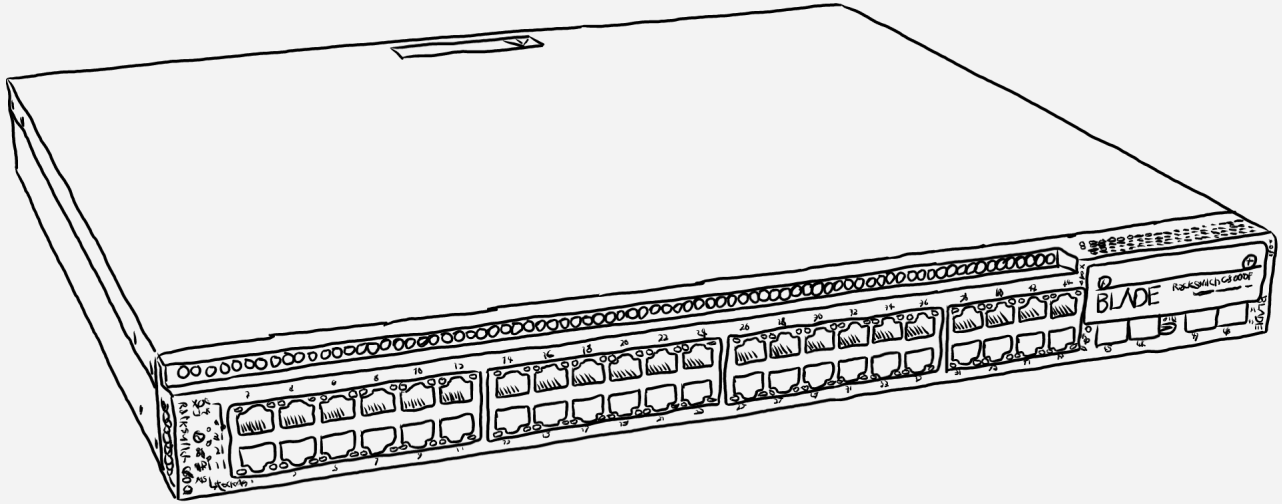
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products 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, zIIPs, 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 (“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.

Agenda

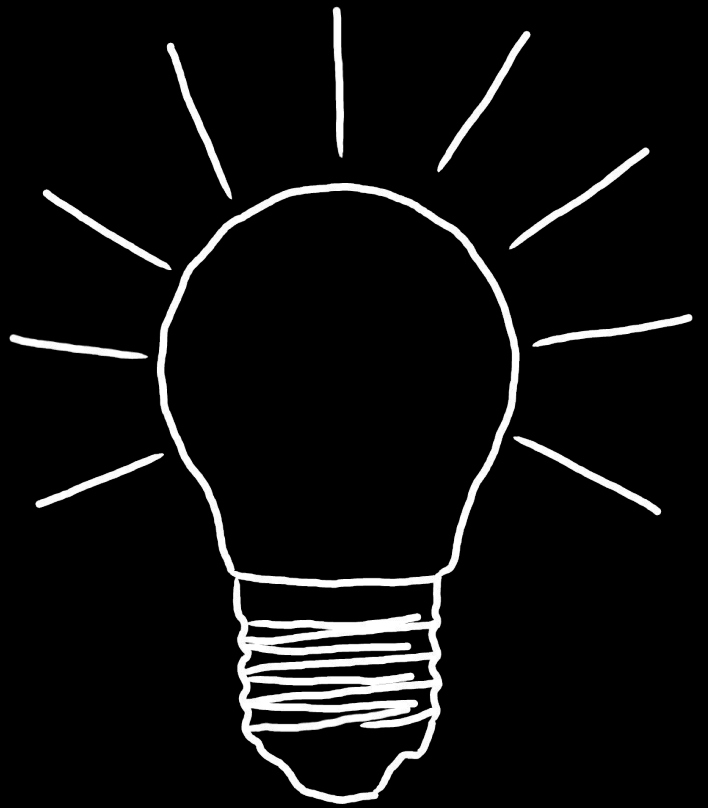
- Basics
- Prerequisites
- Setup & Verification
- Application Enablement
- Monitoring
- Tunables
- Performance
- Summary



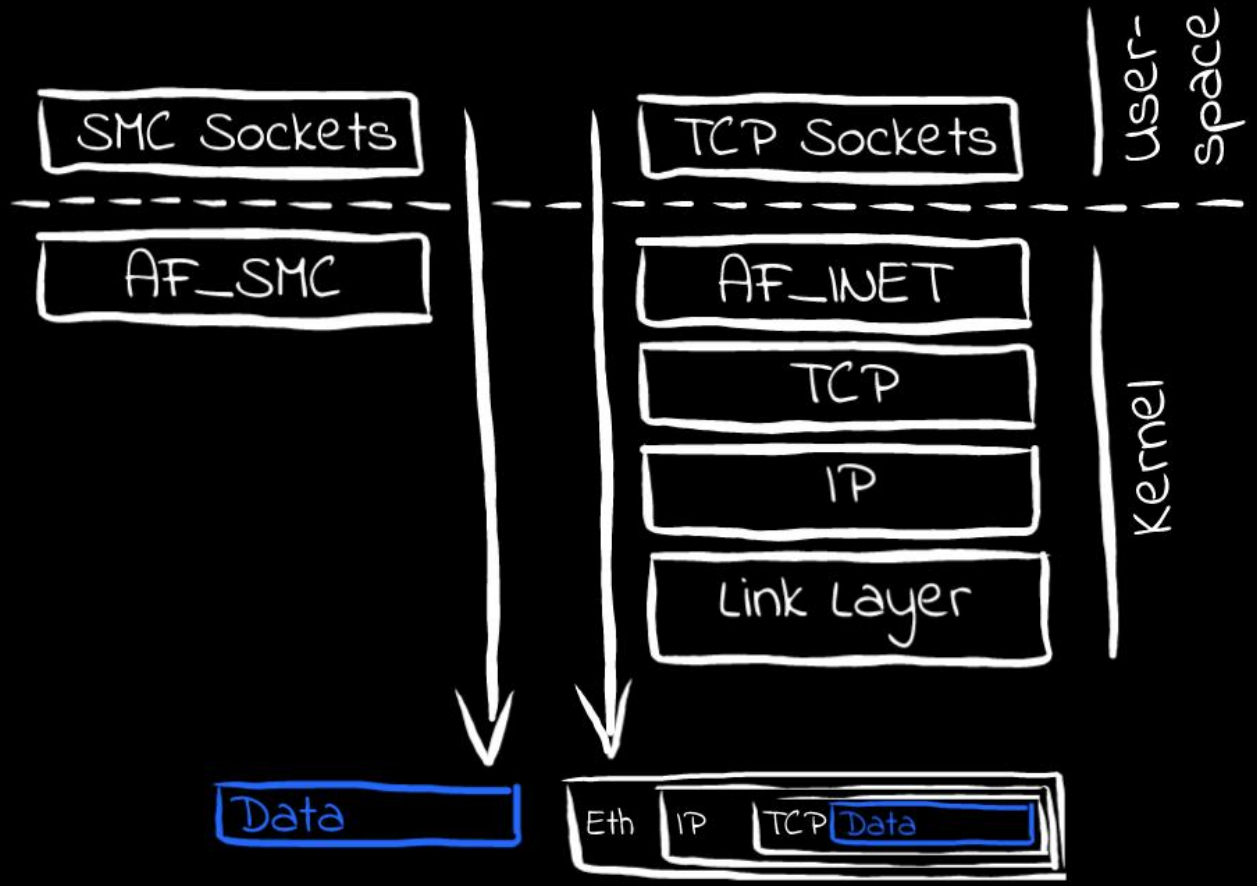
What if we had a networking technology that could provide

- **low latency**
- **high throughput**

and **save CPU cycles** at the same time?



Bypassing the TCP/IP Stack



Performance of SMC-D vs HiperSockets on IBM z15

Benchmark Setup

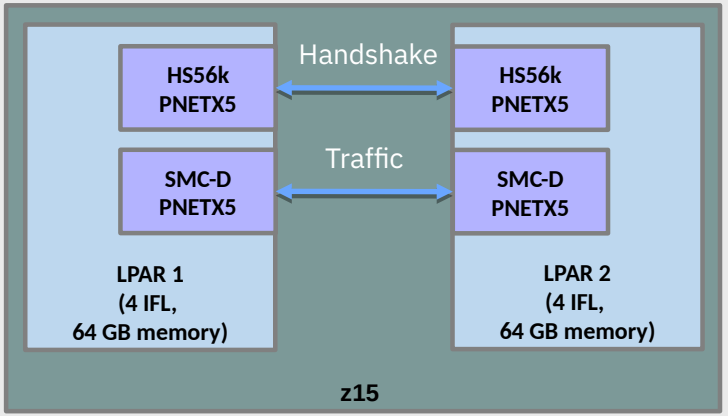
– Ran `uperf` network benchmark with different workload profiles:
  - Highly transactional, medium data sizes: iteratively send 200 bytes of data and receive 1000 bytes of data (client point of view)
  - Transactional, large data sizes: iteratively send 200 bytes of data and receive 30720 bytes of data (client point of view)
  - Streaming writes: continuously write in 30720 byte chunks of data (client point of view)

– Each workload profile was run with 1, 10, 50, and 250 parallel connections

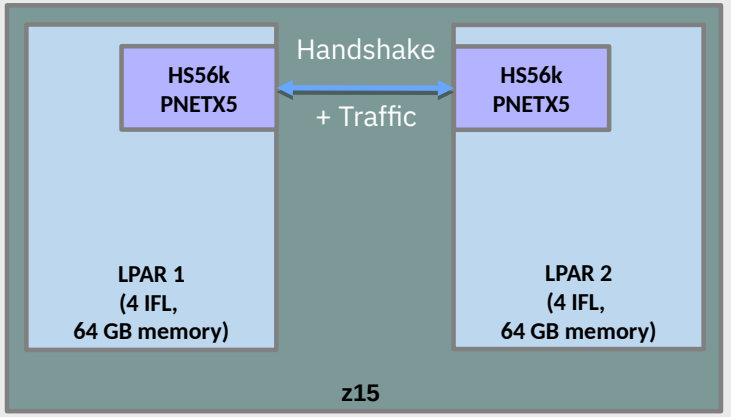
System Stack

– z15
  - 2 LPARs, each with 4 dedicated IFLs, 64 GB memory, 40 GB DASD storage, running SLES 12 SP4 with SMT enabled
  - IFLs of both LPARs were placed on the same chip
  - HiperSockets configured with 56k (HS56k) with an MTU size of 57344 B
  - uperf network benchmark

SMC-D Setup

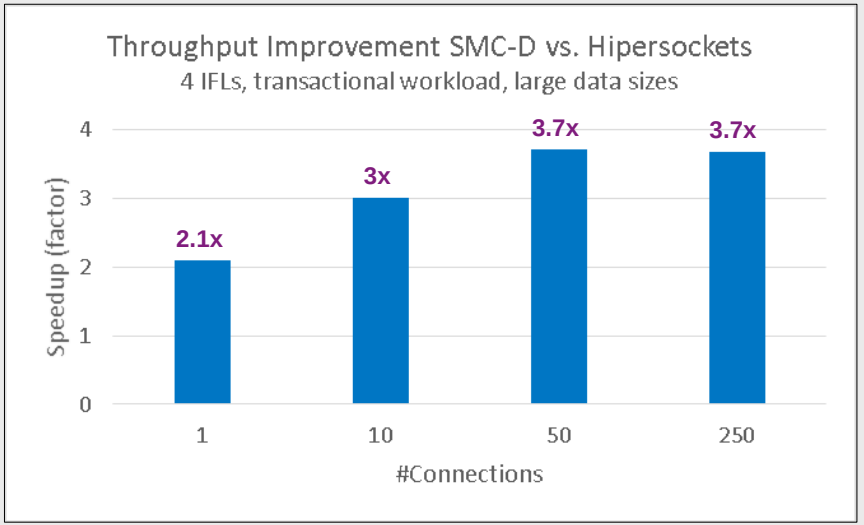


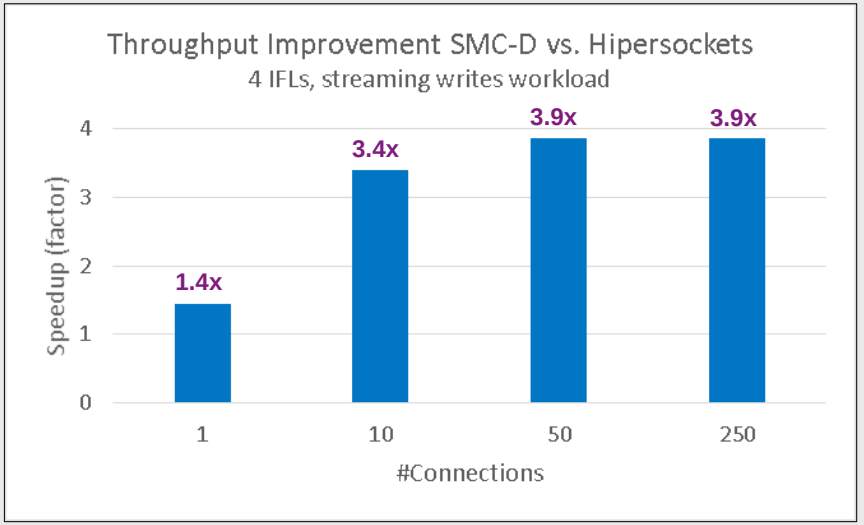
HiperSocket Setup



Performance of SMC-D vs HiperSockets

SMC-D delivers up to 3.9X more throughput between z15 LPARs compared to using Hipersockets

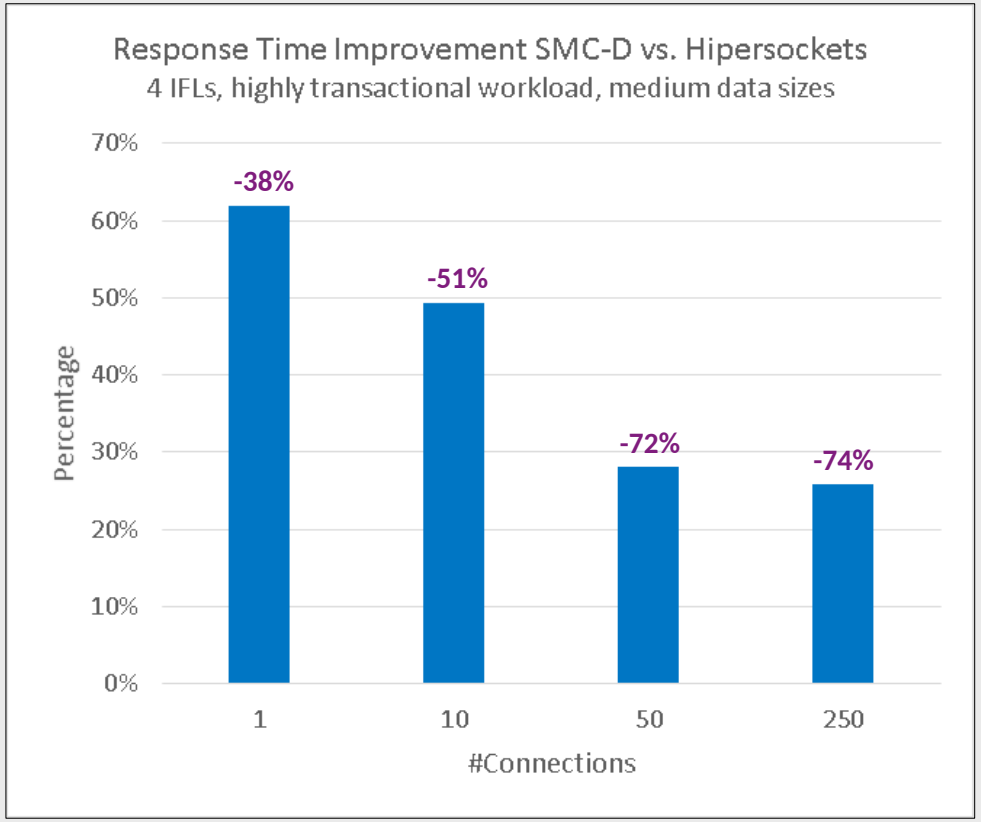




DISCLAIMER: Performance results based on IBM internal tests running uperf (downloaded from https://github.com/uperf/uperf/tree/09fbbdb93e4f0e6569bd532ffd5a4d5969d3eb84) to measure network performance between z15 LPARs. Results may vary. z15 configuration: 2 LPARs, each with 4 dedicated IFLs, 64 GB memory, SLES 12 SP4 (SMT mode) running uperf with different network workload profiles. IFLs of both LPARs were placed on the same chip.

Performance of SMC-D vs HiperSockets on IBM z15

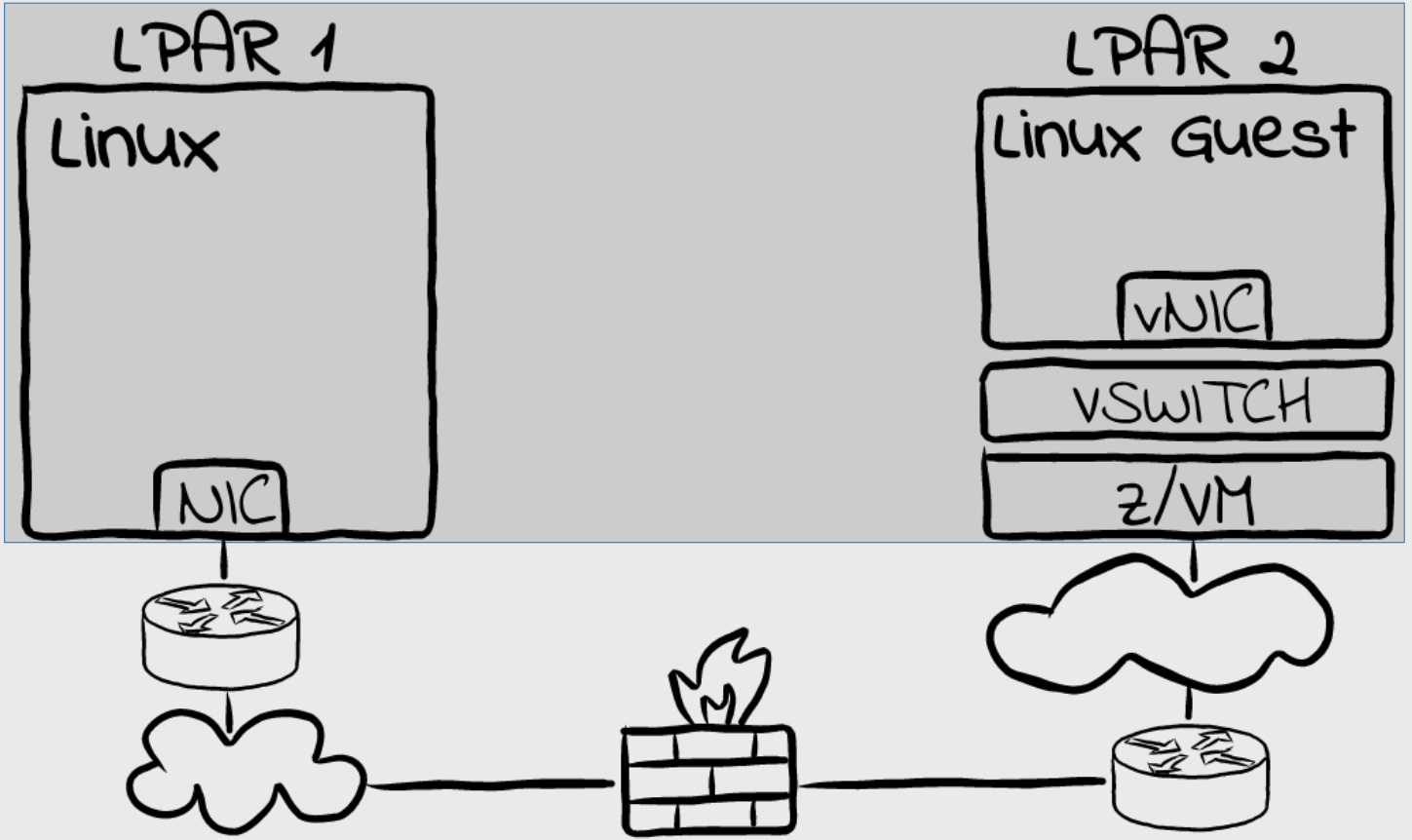
SMC-D delivers up to 74% shorter response time between z15 LPARs compared to using HiperSockets



DISCLAIMER: Performance results based on IBM internal tests running uperf (downloaded from https://github.com/uperf/uperf/tree/09fbbdb93e4f0e6569bd532ffd5a4d5969d3eb84) to measure network performance between z15 LPARs. Results may vary. z15 configuration: 2 LPARs, each with 4 dedicated IFLs, 64 GB memory, SLES 12 SP4 (SMT mode) running uperf with different network workload profiles. IFLs of both LPARs were placed on the same chip.

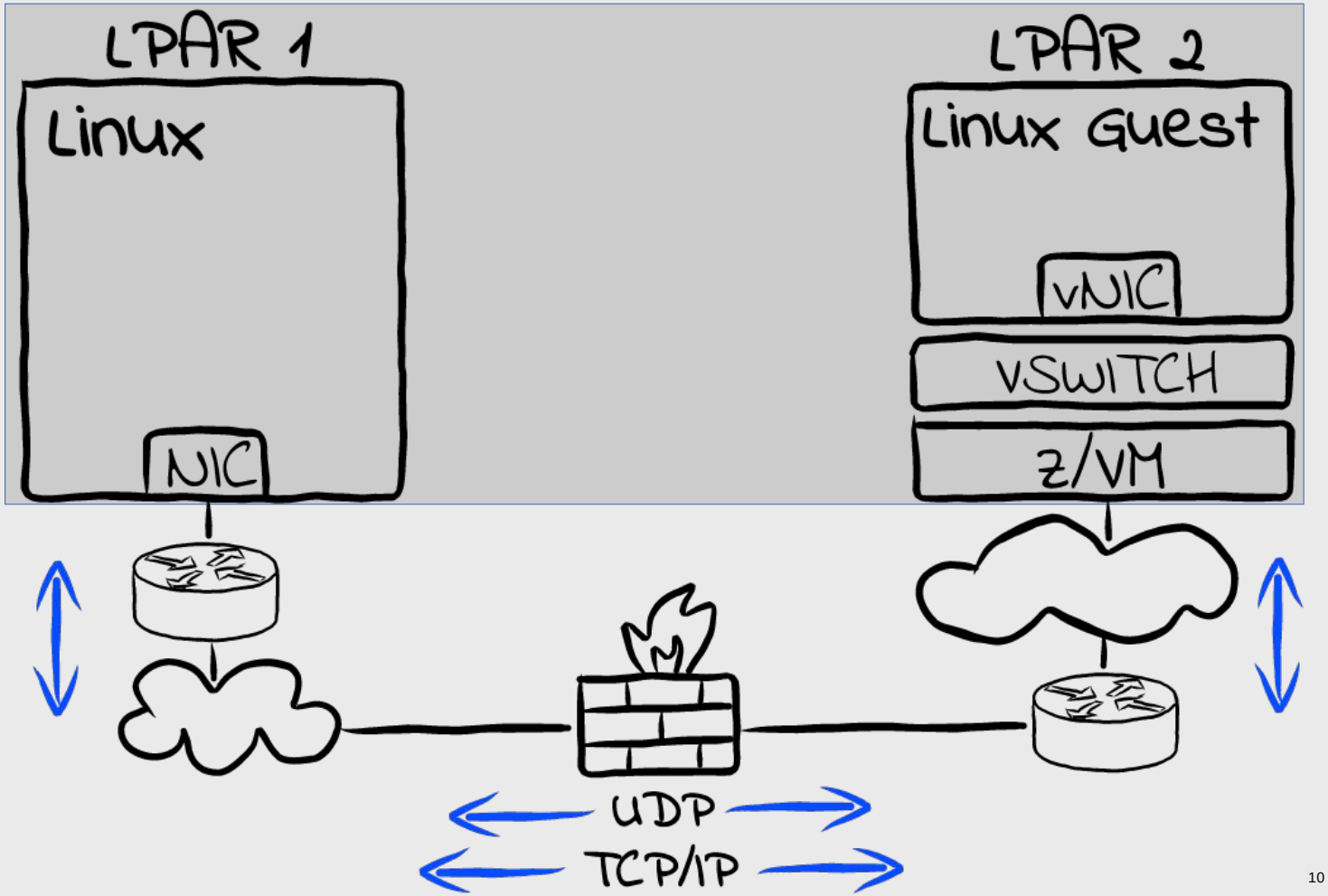
Deployment Scenario

Could be any networking topology as long as both LPARs are located on the same CPC



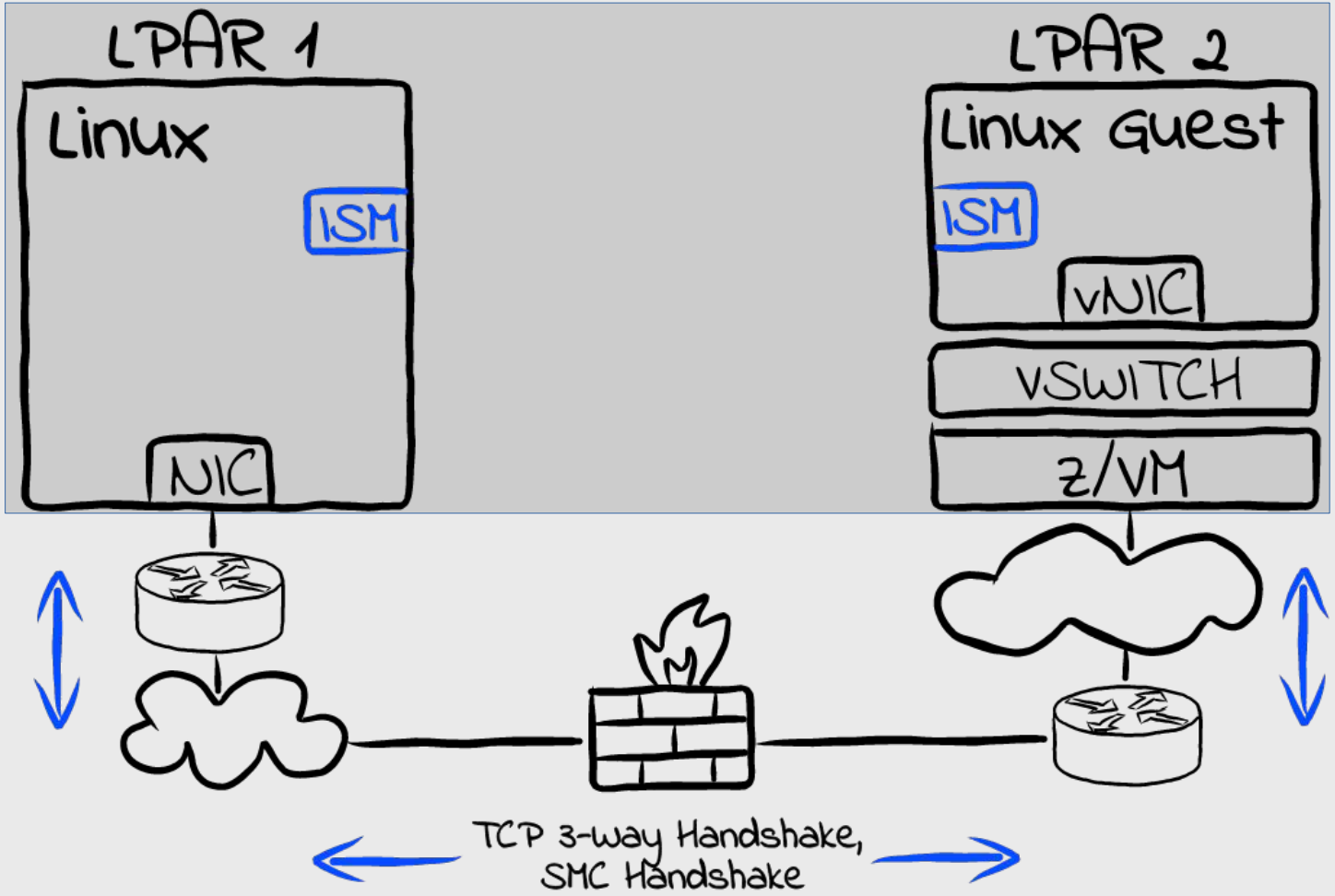
Traffic Flows

HiperSockets might be an obvious choice, but security policies often mandate traffic to pass an external firewall



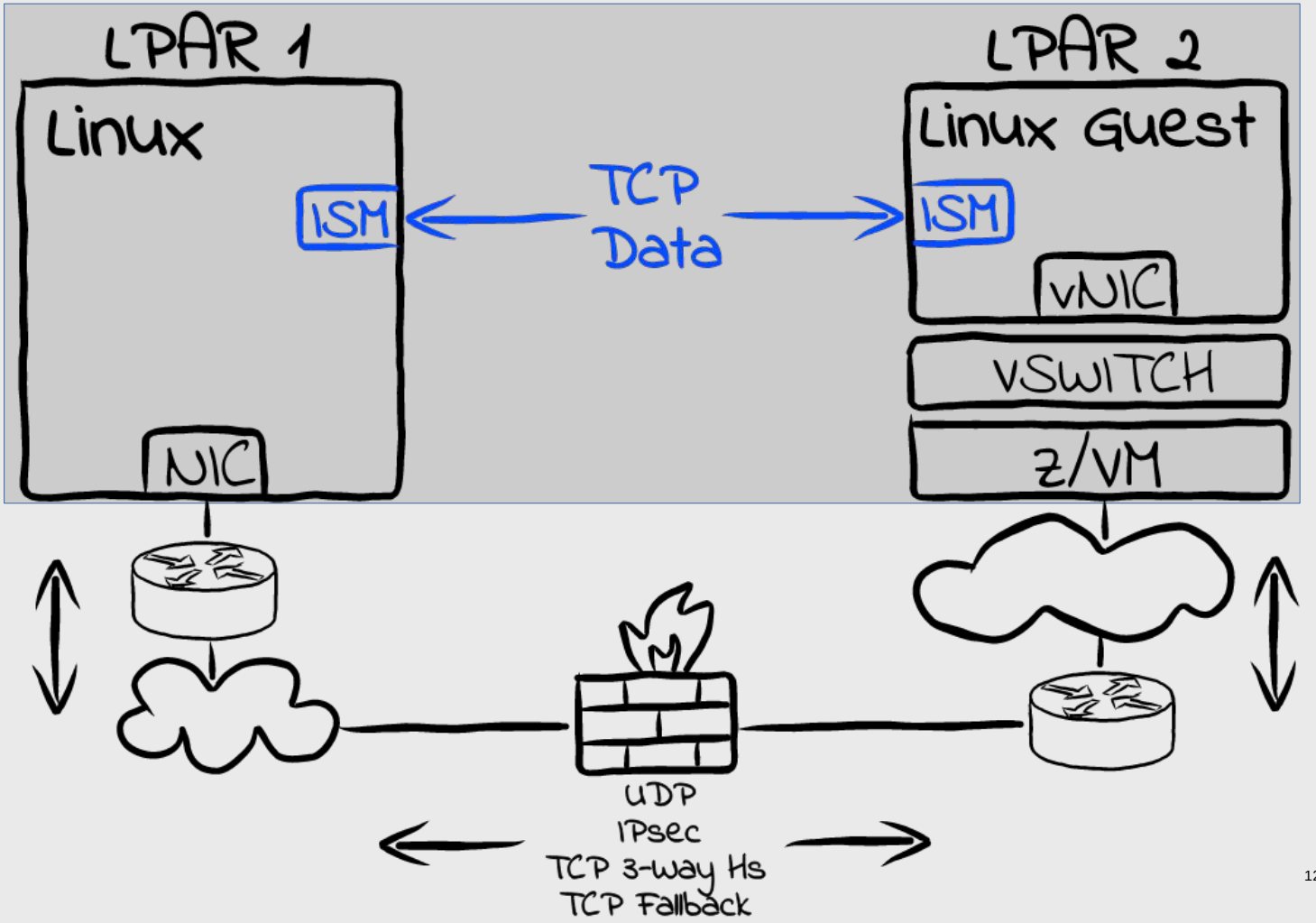
Establishing Connections

- **TCP 3-way handshake is followed by an extra SMC-specific handshake for each new connection**
- **Honors firewall rules!**
- **Overhead is minor, but to be considered for short-lived connections**



Data Flow

- **Once established, TCP data is transmitted through memory-to-memory copy via ISM devices**
- **Uneligible traffic takes the “*detour*” ⇒ Regular connec-tivity still needed**



Summary

- **SMC-D accelerates TCP LPAR-to-LPAR traffic by using memory-to-memory copies, bypassing**
  1) the TCP/IP stack
  2) the connecting networking fabric
- **At the same time,**
  - SMC-Dv2 works for *any* network topology
  - SMC-D honors security policies

Hardware Prerequisites

- **IBM Z hardware requirements**
  - IBM z15 or LinuxONE III
  - Classic mode only (i.e. DPM not supported)
- ***Internal Shared Memory* (ISM) devices**
  - *Virtual* PCI network adapter of VCHID type ISM
  - 32 ISM VCHIDs per CPC, 255 FIDs per VCHID ⇒ 8K FIDs per CPC total)
  - I.e. maximum of 255 virtual servers communicating over same ISM VCHID
  - Each ISM device currently handling up to 1,920 connections
  - Assign multiple ISM devices to increase connection limit

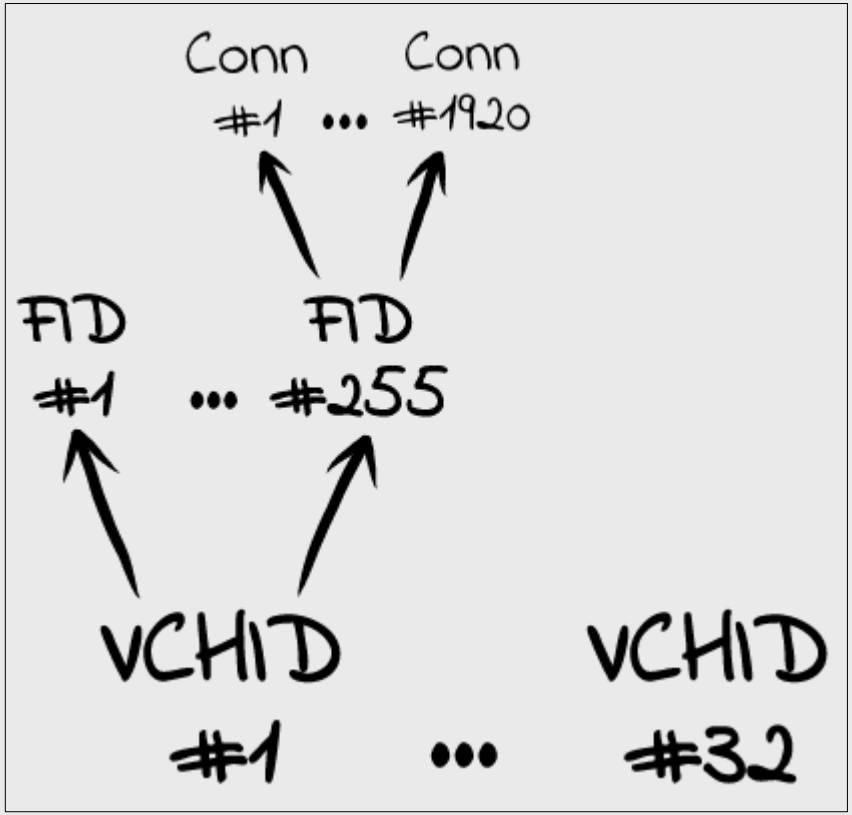


Fig.1: *Relationship between VCHIDs, FIDs and connections*

Software Prerequisites

- **Software**
  - **Supported Linux minimum distribution levels**
    - Ubuntu 21.04
    - RHEL 8.4
    - SLES 15 SP3
  - ***smc-tools*** installed – via Linux distribution, or from https://github.com/ibm-s390-linux/smc-tools
  - **z/OS**:
    - IBM z/OS V2R4 (via APAR) or later
    - Enable SEID in z/OS!
      - Disabled by default
      - See SYSTEMEID in TCP/IP profile GLOBALCONFIG for further details

- **Environments**
  - **Support status:**
    - **LPAR** yes
    - **z/VM guests** yes
    - **KVM guests** WIP
    - **Containers** WIP
  - **Note**: SMC does not support *Live Guest Migration* (LGM) on z/VM and KVM hypervisors.

Setup

- **SMC-Dv2 ISM device eligibility:**
  - (***recommended***) ISM devices without PNET ID
  - ISM devices with PNET ID matched by any networking interface (SMC-Dv1 compatibility)

- **ISM Device Setup**
  - Assign an ISM Device *without* a PNET ID
  - **`smc_rnics`**: Hotplug ISM devices, verify ISM presence, and check PNET IDs

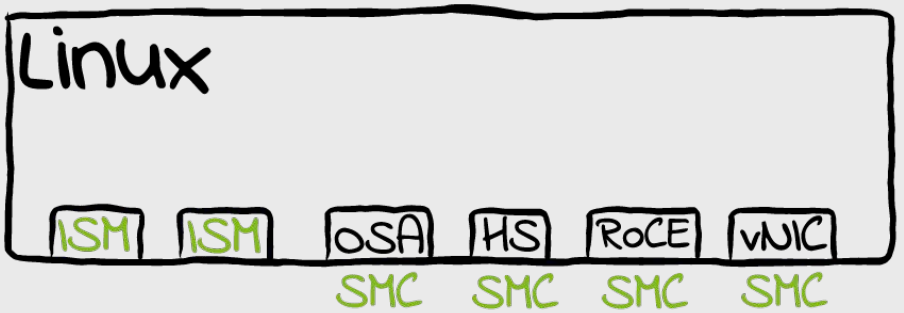


Fig.1: **Any** *interface is enabled for SMC-Dv2 – no further per-interface setup required!*

```
root:~# smc_rnics -a
FID  Power  PCI_ID        PCHID  Type           Port  PNET_ID
-------------------------------------------------------------
 80  1      0000:00:00.0  07c0   ISM            n/a   n/a
 81  0

root:~# smc_rnics -e 81
root:~# smc_rnics
FID  Power  PCI_ID        PCHID  Type           Port  PNET_ID
-------------------------------------------------------------
 80  1      0000:00:00.0  07c0   ISM            n/a   n/a
 81  1      0001:00:00.0  07c0   ISM            n/a   NET2
```

Works under all circumstances!

Would require interface with identical PNET ID!

Verification

- **`smcd info`**[1]: Verify hardware and software support

```
root:~# smcd info
   Kernel Capabilities
   SMC Version:      2.0
   SMC Hostname:     tux
   SMC-D Features:   v1 v2
   SMC-R Features:   v1

   Hardware Capabilities
   SEID:             IBM-SYSZ-ISMSEID0000...
   ISM:              v1 v2
   RoCE:             v1
```

- **`smc_chk`**[1]: Live-test connectivity (will also report local setup issues)

```
root@t83lp76:~# ./smc-tools/smc_chk -S &
Server started on port 37373

root@t83lp76:~# smc_chk -C 127.0.0.1 -p 37373
Test with target IP 127.0.0.1 and port 37373
  Live test (SMC-D and SMC-R)
    Success, using SMC-D

root:~# smc_chk -C 192.168.5.47 -p 23
  Live test (SMC-D and SMC-R)
    Failed  (TCP fallback), reasons:
      Client:  0x03010000  Peer does not support SMC
```

Live test with local server, also provided by `smc_chk`

Live test with ssh server on remote

[1] *smc-tools* v1.5 or later

Application Enablement: 2 Ways to Enable Applications

1) Use pre-load library `libsmc-preload.so`

- Provided by *smc-tools*
- Intercepts existing applications' `socket()` calls
- Two ways to enable:
  a) Use `smc_run` (recommended)

```
root:~# smc_run <my_application>
```

  b) Enable through environment variable:

```
root:~# export LD_PRELOAD=libsmc-\
                          preload.so
```

- **Note**: Will not work with statically linked applications! (rare case)

2) Alternative: Re-compile the application

- SMC implemented as separate address family `AF_SMC`.
- In applications' `socket()` calls, replace `AF_INET` with `AF_SMC`, i.e.:

```
int s, ipv6 = 0;

s = socket(AF_SMC, SOCK_STREAM, ipv6);
```

- Unlikely to happen with users' applications

Application Enablement with Preload Library

Three levels of enablement to chose from:

a) **Per Application**: Use `smc_run`

or

b) **Per User**: Set `LD_PRELOAD` in the profile of the user ID that starts the respective processes, e.g. the DB2 instance owner:

```
root:~# echo "export LD_PRELOAD=\
        libsmc-preload.so" >> ~/.profile
```

or

c) **OS Global**: Use `/etc/ld.so.preload` to enable the entire system:

```
root:~# cat /etc/ld.so.preload
libsmc-preload.so
```

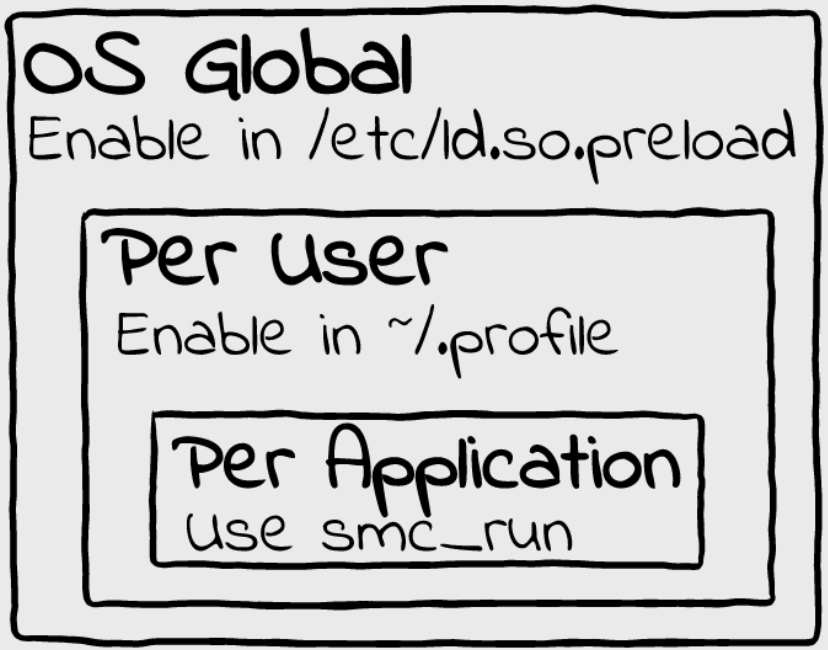


Fig.1: *The different levels of SMC enablement*

Enabling Unruly Applications

- Some applications routinely clear environment variables
- I.e. application enablement using preload library via environment variable will not work
- However, most applications provide means to pass on environment variables, still.
- E.g. DB2 requires registration of environment variables through `db2set` command.

```
root:~# export LD_PRELOAD=libsmc-preload.so
root:~# db2set -i db2inst1 DB2ENVLIST="LD_LIBRARY_PATH LD_PRELOAD"
root:~# smc_run db2start
```

Enabling Container Workloads

- **Containers run in isolated networks**
  ⇒ SMC-Dv2 is prerequisite for enablement

- **Approach**
  - Enable base image for SMC-Dv2 by
    - install *smc-tools* package
    - globally enable preload library (unless you want a “fancy” solution)
  - **Deployment**
    - make ISM device available to containers
    - monitor correct operation

    This is (currently!) the tricky part!

Monitoring Connections

- Use command **`smcss`**:
  - Monitor SMC-enablement socket status (see column "`Mode`")
  - Consult `smcss` man page for error codes of fallback connections

```
root:~# smcss -a
State   UID    Inode   Local Address      Foreign Address    Intf  Mode
ACTIVE  20000  115762  192.168.5.8:6059   192.168.5.49:3220  0000  SMCD
ACTIVE  20000  115482  192.168.5.8:2183   192.168.5.47:8973  0000  TCP   0x03010000
```

```
Linux error codes:

0x01010000   Out of memory

0x02010000   Timeout while waiting for confirm link message over RDMA device

0x02020000   Timeout while waiting for RDMA device to be added

0x03000000   Configuration error

0x03010000   Peer does not support SMC

0x03020000   Connection uses IPsec
```

Fig.1: *`smcss` man page*

Statistics[1]

- **Statistics provide summary of SMC-D enabled connections (successful and fallback)**
- **Can serve as basis for further optimizations to improve performance**
- **Supports data export in JSON format for further processing**

```
root:~# smcd stats
SMC-D Connections Summary
  Total connections handled        152730
  SMC connections                  152730
  Handshake errors                      0
  Avg requests per SMC conn         813.1
  TCP fallback                          0

RX Stats
  Data transmitted (Bytes)   270311225427 (270.3G)
  Total requests                 61619256
  Buffer full                      114746 (0.19%)
            8KB    16KB    32KB    64KB   128KB   256KB   512KB  >512KB
  Bufs        0       2     140  2.103K       0       2       0  95.97K
  Reqs   54.07M       3  7.552M       0       0       0       0       0

TX Stats
  Data transmitted (Bytes)   271274963896 (271.3G)
  Total requests                 62565728
  Buffer full                           0 (0.00%)
  Buffer full(remote)               90038 (0.14%)
  Buffer too small                      0 (0.00%)
  Buffer too small(remote)              0 (0.00%)
            8KB    16KB    32KB    64KB   128KB   256KB   512KB  >512KB
  Bufs        0  2.384K     142       0       0       2       0       0
  Reqs   54.92M       3  7.552M       0       0       0       0       0

Extras
  Special socket calls                  0
```

Should be “sufficiently high” *for efficient SMC usage*

[1] *smc-tools* v1.6

Buffer Usage

- **Buffers store raw application data only – no headers included**
- **Available buffer sizes: 8KB, 16KB,..., 1MB**
- **Increase buffer sizes to reduce risk of buffer full conditions**
- **Use `smcd stats` output to check on buffer usage:**

```
root:~# smcd stats
[...]
RX Stats
  Data transmitted (Bytes)   552475545397 (552.5G)
  Total requests                 92515597
  Buffer full                     1783848 (1.93%)
            8KB    16KB    32KB    64KB   128KB   256KB   512KB  >512KB
  Bufs        0       2     220  2.409K       0       2       0  97.03K
  Reqs   76.63M  1.661M  12.51M  1.332M       0       0       0       0

TX Stats
  Data transmitted (Bytes)   553440720258 (553.4G)
  Total requests                 99144120
  Buffer full                           4 (0.00%)
  Buffer full(remote)            16981353 (17.13%)
  Buffer too small                1775733 (1.79%)
  Buffer too small(remote)        1773141 (1.79%)
[...]
```

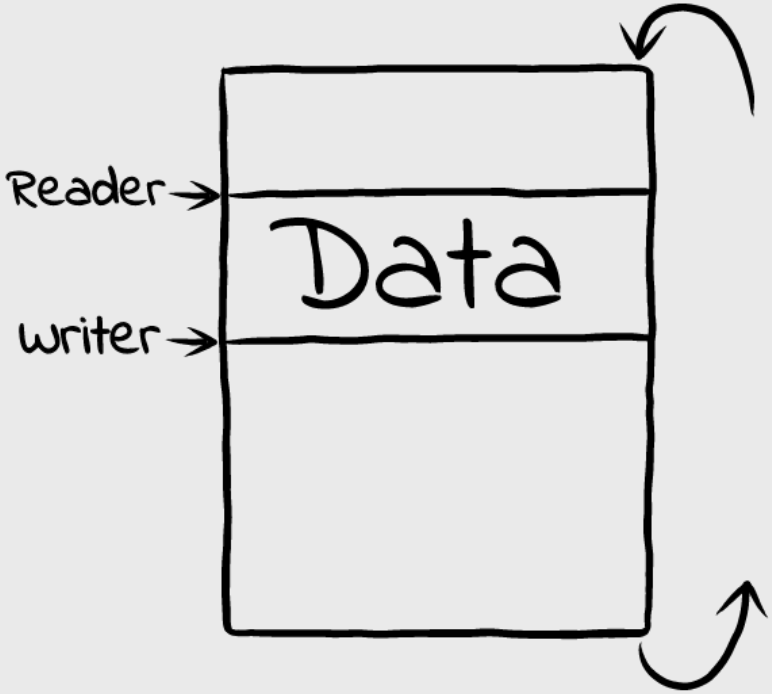


Fig.1: *SMC-D Buffer*

Pathologic mismatches between buffer and request sizes may indicate source of inefficiencies

Requesting specific Buffer Sizes[1]

- **Preload-library** via environment variables, or **`smc_run`**[1]:

```
# using smc_run
root:~# smc_run -r 1M -t 128K ./foo

# same settings using env variables for preload library
root:~# export SMC_RCVBUF=1M
root:~# export SMC_SNDBUF=128K
```

- **Global setting** (affects *all* connections) via `sysctl`:

```
# receive buffers (aka RMBEs)
root:~# sysctl -w /proc/sys/net/core/rmem_max=1048576
root:~# sysctl -w net.ipv4.tcp_rmem="4096 1048576 6291456"

# send buffers
root:~# sysctl -w /proc/sys/net/core/wmem_max=1048576
root:~# sysctl -w net.ipv4.tcp_wmem="4096 1048576 4194304"
```

- **Notes**
  - Applications might override the requested buffer sizes
  - Memory fragmentation might prevent huge buffers in long-running systems
  - Check statistics for actual buffer sizes

[1] *smc-tools* v1.6

Summary

Key Attributes

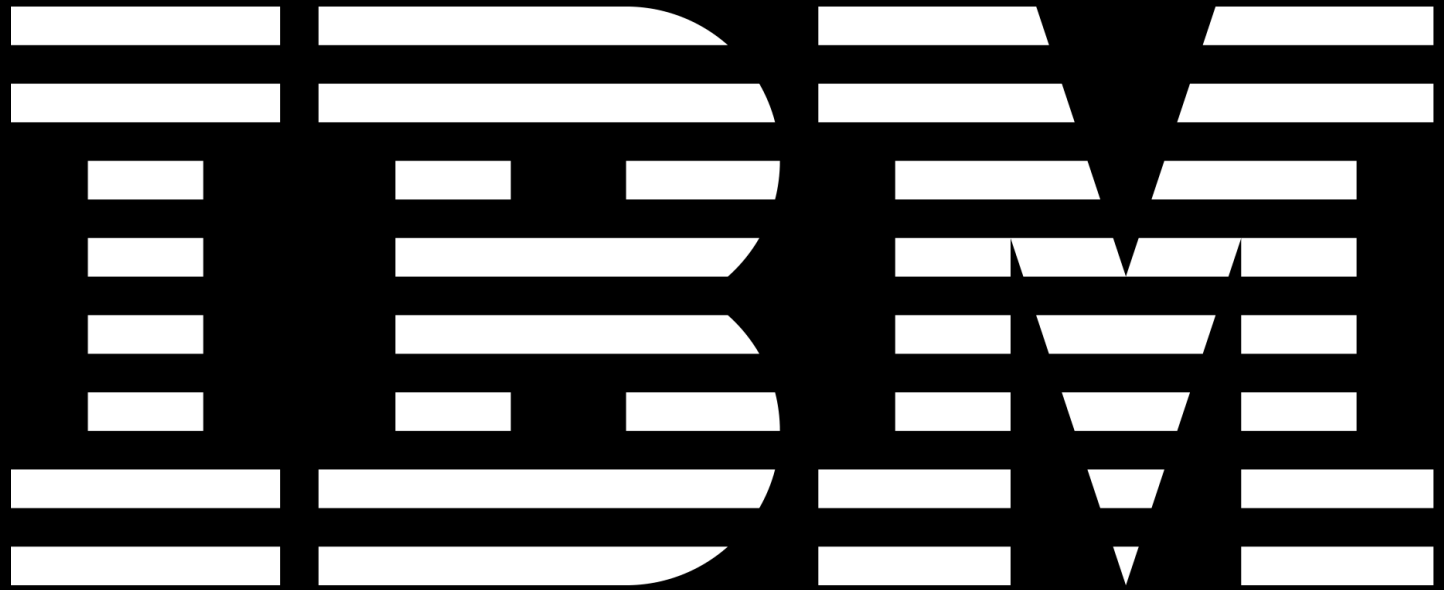
- Supports peers in different IP subnets
- Easy HW setup: Add ISM device – done!
- Massive performance benefits as compared to HiperSockets – gets even better when compared to regular NICs!
- Transparent to (TCP socket based) applications
- Preserves existing network addressing-based security models
- Transparent to network components such as channel bonding and load balancers

Typical Workloads To Benefit

- *Transaction-oriented* / *latency-sensitive*
- *bulk data streaming*, e.g. when running backups
- Huge amounts of concurrent connections

References

- **SMC for Linux on Z**
  http://linux-on-z.blogspot.com/p/smc-for-linux-on-ibm-z.html
- ***smc-tools* Homepage**
  https://github.com/ibm-s390-linux/smc-tools
- **RFC7609 (SMC-R)**
  https://tools.ietf.org/html/rfc7609
- **Linux on Z Documentation**
  https://www.ibm.com/docs/en/linux-on-systems?topic=linux-z-linuxone
- **Webcasts**
  http://ibm.biz/Linux-on-IBMZ-LinuxONE-Webcasts
- **Blogs**
  - **Linux on z distributions new**
    http://linuxmain.blogspot.com/
  - **Linux on Z latest development news**
    http://linux-on-z.blogspot.com/
  - **KVM on Z**
    http://kvmonz.blogspot.com/



Backup

Comparison

Feature	SMC-Dv2	SMC-Dv1	SMC-Rv1
Intra-CPC	yes	yes	yes
Cross-CPC	no	no	yes
Cross-IP subnet	**yes**	no	no
(R)DMA Device	ISM	ISM	RoCE
Bus used	-	-	PCI
PNET ID Definition	**Not required**	IOCDS, or `smc_pnet`	IOCDS, or `smc_pnet`
Failover	N/A	N/A	yes
Upstream Status	Linux kernel 5.10 or later	Linux kernel 4.19 or later	Linux kernel 4.18 or later

Migrating from SMC-Dv1

- **SMC-Dv1 setups require compatibility patches for SMC-Dv2 interoperability**
  - RHEL 8.1, Linux kernel 4.18.0-147.27.1
  - RHEL 8.2, Linux kernel 4.18.0-193.28.1
  - RHEL 8.3, Linux kernel 4.18.0-228
  - SLES 12 SP5, Linux kernel 4.12.14-122.41.1
  - SLES 15 SP1, Linux kernel 4.12.14-197.61.1
  - SLES 15 SP2, Linux kernel 5.3.18-24.9.1
  - Ubuntu 20.04, Linux kernel 5.4.0-45.49
- **SMC-Dv1 compatibility mode in SMC-Dv2 only available on interfaces and ISM devices with matching PNET IDs**

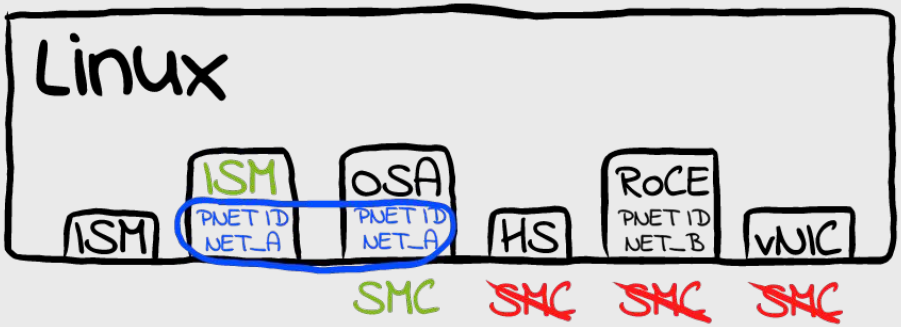


Fig.1: *SMC-Dv1: Only interfaces with matching PNET IDs are enabled for SMC-D*

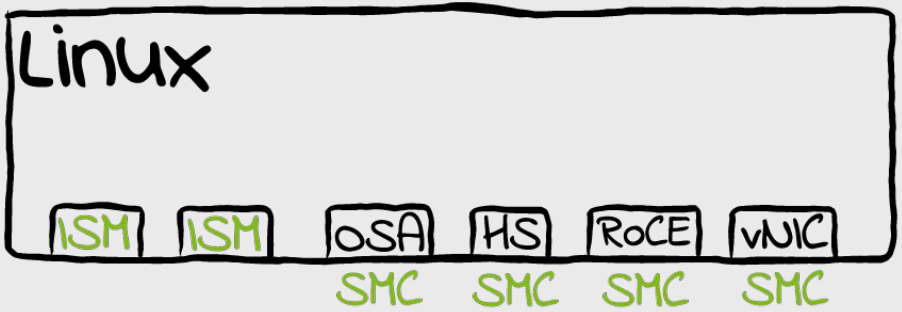


Fig.2: *SMC-Dv2: Any interface is enabled for SMC-D – no PNET ID required!*

smc-tools Package Overview

- **Current version**: v1.5
- Homepage: https://github.com/ibm-s390-linux/smc-tools
- *smc-tools* provides the following commands:
  - **`smc_pnet`**: Not required for SMC-Dv2.
  - **`smc_run`**: Enable binary applications to use SMC.
  - **`smcss`**: Information about SMC-enabled sockets and link groups. Includes information on SMC mode used, as well as TCP fallbacks
  - **`smc_rnics`** (v1.2 or later): List, hotplug and hot-unplug PCI (R)DMA devices
  - **`smcd`** (v1.4 or later): Information on ISM devices, soft- and hardware support levels, usage
  - **`smc_chk`** (v1.5 or later): SMC support diagnostics
  - **`smc_dbg`** (v1.2 or later): Collect debugging information