Linux on System z Performance Update - Part 2 Networking and Crypto

Mario Held
IBM Research & Development, Germany

August 28, 2009
Session Number 2192

Trademarks

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

DB2*
DB2 Connect
DB2 Universal Database
e-business logo
GDPS*
Geographically Dispersed Parallel Sysplex
HyperSwap
IBM*
IBM eServer
IBM logo*
Parallel Sysplex*
System z
Tivoli*
VM/ESA*
WebSphere*
z/OS*
z/VM*
zSeries*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Intel is a registered trademark of the Intel Corporation in the United States, other countries or both.

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

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

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

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective 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.

Agenda

- **Network**
- Cryptographic hardware support

General Network Performance (1)

- Which connectivity to use:
  - External connectivity:
    - LPAR: 10 GbE cards
    - z/VM: VSWITCH with 10GbE card(s) attached
    - z/VM: For maximum throughput and minimal CPU utilization attach OSA directly to Linux guest
  - Internal connectivity:
    - LPAR: HiperSockets for LPAR-LPAR communication
    - z/VM: VSWITCH for guest-guest communication
- For the highly utilized network
  - use z/VM VSWITCH with link aggregation
  - use channel bonding

General Network Performance (2)

- If network performance problems are observed, the buffer count can be increased up to 128
  - The default of 16 buffers limits the throughput of a HiperSockets connection with 10 parallel sessions.
  - Check current buffer count with **`lsqeth -p`** command
  - A buffer count of 128 leads to 8MB memory consumption.
    - One buffer consists of 16x4KB pages which yields 64KB, so 128x64KB=8MB.
- Set the inbound buffer count in the appropriate config file:
  - SUSE SLES10:
    in **`/etc/sysconfig/hardware/hwcfg-qeth-bus-ccw-0.0.F200`**
    add **`QETH_OPTIONS="buffer_count=128"`**
  - SUSE SLES11:
    in **`/etc/udev/rules.d/51-qeth-0.0.f200.rules`**
    add **`ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.f200", ATTR{buffer_count}="128"`**
  - Red Hat:
    in **`/etc/sysconfig/network-scripts/ifcfg-eth0`**
    add **`OPTIONS="buffer_count=128"`**

General Network Performance (3)

- Consider to switch on priority queueing if an OSA Express adapter in QDIO mode is shared amongst several LPARs. How to achieve:
  - SUSE SLES10:
    in `/etc/sysconfig/hardware/hwcfg-qeth-bus-ccw-0.0.F200`
    add `QETH_OPTIONS="priority_queueing=no_prio_queueing:0"`
  - SUSE SLES11:
    in `/etc/udev/rules.d/51-qeth-0.0.f200.rules`
    add `ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.f200", ATTR{priority_queueing}="no_prio_queueing:0"`
  - Red Hat:
    in `/etc/sysconfig/network-scripts/ifcfg-eth0`
    add `OPTIONS="priority_queueing=no_prio_queueing:0"`
- Note: Priority queueing on one LPAR may impact the performance on all other LPARs sharing the same OSA card.

General Network Performance (4)

- Choose your MTU size carefully. Set it to the maximum size supported by all hops on the path to the final destination to avoid fragmentation.
  - Use `tracepath <destination>` to check MTU size
  - If the application sends in chunks of <=1400 bytes, use MTU 1492.
    - 1400 Bytes user data plus protocol overhead.
  - If the application is able to send bigger chunks, use MTU 8992.
    - Sending packets > 1400 with MTU 8992 will increase throughput and save CPU cycles.
- TCP uses the MTU size for the window size calculation, not the actual application send size.
- For VSWITCH, MTU 8992 is recommended.
  - Synchronous operation, SIGA required for every packet.
  - No packing like normal OSA cards.

General Network Performance (5)

- Following system wide sysctl settings can be changed temporarily by the sysctl command or permanently in the appropriate config file:

  ```
  /etc/sysctl.conf
  ```

- Set the device queue length from the default of 1000 to at least 2500 using sysctl:

  ```
  sysctl -w net.core.netdev_max_backlog = 2500
  ```

- Adapt the inbound and outbound window size to suit the workload
  - Recommended values for OSA devices:

    ```
    sysctl -w net.ipv4.tcp_wmem="4096 16384 131072
    sysctl -w net.ipv4.tcp_rmem="4096 87380 174760
    ```

  - System wide window size applies to all network devices
    - Applications can use setsockopt to adjust the window size
    - Has no impact on other network devices
  - More than ten parallel sessions benefit from recommended default and maximum window sizes.
  - A bigger window size can be advantageous for up to five parallel sessions.

General Network Performance (6)

- By default, numerous daemons are started that might be unnecessary.
  - Check whether selinux and auditing are required for your business. If not, switch them off.

```
dmesg | grep -i "selinux"
chkconfig --list | grep -i "auditd"
```

In /etc/zipl.conf

```
parameters="audit_enable=0 audit=0
  audit_debug=0selinux=0 ..."
chkconfig --set auditd off
```

  - Check which daemons are running

```
chkconfig --list
```

  - Examples of daemons which are not required under Linux on System z:
    - alsasound
    - avahi-daemon
    - resmgr
    - postfix

General Network Performance (7)

- A controlled environment without an external network connection (“network in a box”) usually doesn't require a firewall.
  - The Linux firewall (iptables) **serializes the network** traffic because it can utilize just one CPU.
  - If there is no business need, switch off the firewall (iptables) which is enabled by default
  - For example in SLES distributions

```
chkconfig -set SuSEfirewall2_init off
chkconfig -set SuSEfirewall2_setup off
chkconfig -set iptables off
```

Networking - HiperSockets Linux to z/OS, recommendations for z/OS and Linux (1)

- Frame size and MTU size are determined by the chparm parameter of the IOCDS
  - MTU size = frame size – 8KB

chparm	frame size	MTU
00	16KB	8KB
40	24KB	16KB
80	40KB	32KB
C0	64KB	56KB

- Select the MTU size to suit the workload. If the application is mostly sending packets < 8KB, an MTU size of 8KB is sufficient.
- If the application is capable of sending big packets, a larger MTU size will increase throughput and save CPU cycles.
- MTU size 56KB is recommended only for streaming workloads with packets > 32KB.

Networking - HiperSockets Linux to z/OS, recommendations for z/OS and Linux (2)

- HiperSockets doesn't require checksumming because it is a memory-to-memory operation.
  - Default is sw_checksumming.
  - To save CPU cycles switch checksumming off:

SUSE SLES10:
in `/etc/sysconfig/hardware/hwcfg-qeth-bus-ccw-0.0.F200` add
`QETH_OPTIONS="checksumming=no_checksumming"`

SUSE SLES11:
in `/etc/udev/rules.d/51-qeth-0.0.f200.rules` add

```
ACTION=="add", SUBSYSTEM=="ccwgroup", KERNEL=="0.0.f200",
 ATTR{checksumming}="no_checksumming"
```

Red Hat RHEL5 :
in `/etc/sysconfig/network-scripts/ifcfg-eth0` add
`OPTIONS="checksumming=no_checksumming"`

Networking - HiperSockets Linux to z/OS, recommendations for z/OS

- Always set:

```
TCPCONFIG DELAYACKS
```

- If MTU size is 8KB or 16KB set:

```
TCPCONFIG TCPRCVBUFRSIZE 32768
TCPCONFIG TCPSENDBFRSIZE 32768
```

- If MTU size is 32KB set:

```
TCPCONFIG TCPRCVBUFRSIZE 65536
TCPCONFIG TCPSENDBFRSIZE 65536
```

- If MTU size is 56KB set:

```
TCPCONFIG TCPRCVBUFRSIZE 131072
TCPCONFIG TCPSENDBFRSIZE 131072
```

- For HiperSockets MTU 56KB, the CSM fixed storage limit is important. The default is currently 120 MB and should be adjusted to 250MB if MSGIVT5592I is observed

Networking - HiperSockets Linux to z/OS, recommendations for Linux (1)

- The setting of rmem / wmem under Linux on System z determines the minimum, default and maximum window size which has the same meaning as buffer size under z/OS.
- Linux window size settings are system wide and apply to all network devices.
  - Applications can use setsockopt to adjust the window size individually.
    - Has no impact on other network devices
  - HiperSockets and OSA devices have contradictory demands
    - >10 parallel OSA sessions suffer from a send window size > 32KB
    - The suggested default send/receive window size for HiperSockets 8KB and 16KB are also adequate for OSA devices.

Networking - HiperSockets Linux to z/OS, recommendations for Linux (2)

- As a rule of thumb the default send window size should be twice the MTU size, e.g.:
  - MTU 8KB

```
sysctl -w net.ipv4.tcp_wmem="4096 16384 131072
sysctl -w net.ipv4.tcp_rmem="4096 87380 174760
```

  - MTU 16KB

```
sysctl -w net.ipv4.tcp_wmem="4096 32768 131072
sysctl -w net.ipv4.tcp_rmem="4096 87380 174760
```

  - MTU 32KB

```
sysctl -w net.ipv4.tcp_wmem="4096 65536 131072
sysctl -w net.ipv4.tcp_rmem="4096 87380 174760
```

  - MTU 56KB

```
sysctl -w net.ipv4.tcp_wmem="4096 131072 131072
sysctl -w net.ipv4.tcp_rmem="4096 131072 174760
```

Networking - Linux to z/OS, SAP recommendations

- The SAP Enqueue Server requires a default send window size of 4 * MTU size.
- HiperSockets
  - SAP networking is a transactional type of workload with a packet size < 8KB. HiperSockets MTU 8KB is sufficient.

```
sysctl -w net.ipv4.tcp_wmem="4096 32768 131072
sysctl -w net.ipv4.tcp_rmem="4096 87380 174760
```

- OSA
  - If the SAP Enqueue Server connection is run over an OSA adapter, the MTU size should be 8192. If MTU 8992 is used, the default send window size must be adjusted to 4 * 8992.

Network – benchmark description 1/2

- AWM - IBM Benchmark which simulates network workload
  - All our tests are running with 10 simultaneous connections
  - Transactional Workloads – 2 types
    - RR – request/response
      - *A connection to the server is opened once for a 5 minute time frame*
    - CRR – connect/request/response
      - *A connection is opened and closed for every request/response*
    - RR 200/1000 (Send 200 bytes from client to server and get 1000 bytes response)
      - *Simulating online transactions*
    - RR 200/32k (Send 200 bytes from client to server and get 32kb bytes response)
      - *Simulating website access*
    - CRR 64/8k (Send 64 bytes from client to server and get 8kb bytes response)
      - *Simulating database query*
  - Streaming workloads – 2 types
    - STRP - "stream put" (Send 20m bytes to the server and get 20 bytes response)
    - STRG - "stream get" (Send 20 bytes to the server and get 20m bytes response)
      - *Simulating large file transfers*

Network – benchmark description 2/2

- Connection types
  - OSA 1000Base-T MTU sizes 1492 and 8992
  - OSA 1 Gigabit Ethernet MTU sizes 1492 and 8992
  - OSA 10 Gigabit Ethernet MTU sizes 1492 and 8992
  - HiperSockets MTU size 32k
  - GuestLAN type HiperSockets MTU size 32KB (z/VM only)
  - GuestLAN type QDIO MTU sizes 8992 and 32KB (z/VM only)
  - VSWITCH z/VM guest to guest MTU sizes 1492 and 8992 (z/VM only)
  - OSA 1 Gigabit Ethernet dedicated z/VM guest to Linux in LPAR MTU sizes 1492 and 8992
  - OSA 10 Gigabit Ethernet dedicated z/VM guest to Linux in LPAR MTU sizes 1492 and 8992
  - OSA 1 Gigabit Ethernet VSWITCH z/VM guest to Linux in LPAR MTU sizes 1492 and 8992
  - OSA 10 Gigabit Ethernet VSWITCH z/VM guest to Linux in LPAR MTU sizes 1492 and 8992

Database query

SLES10 SP2 / z10

200 byte request

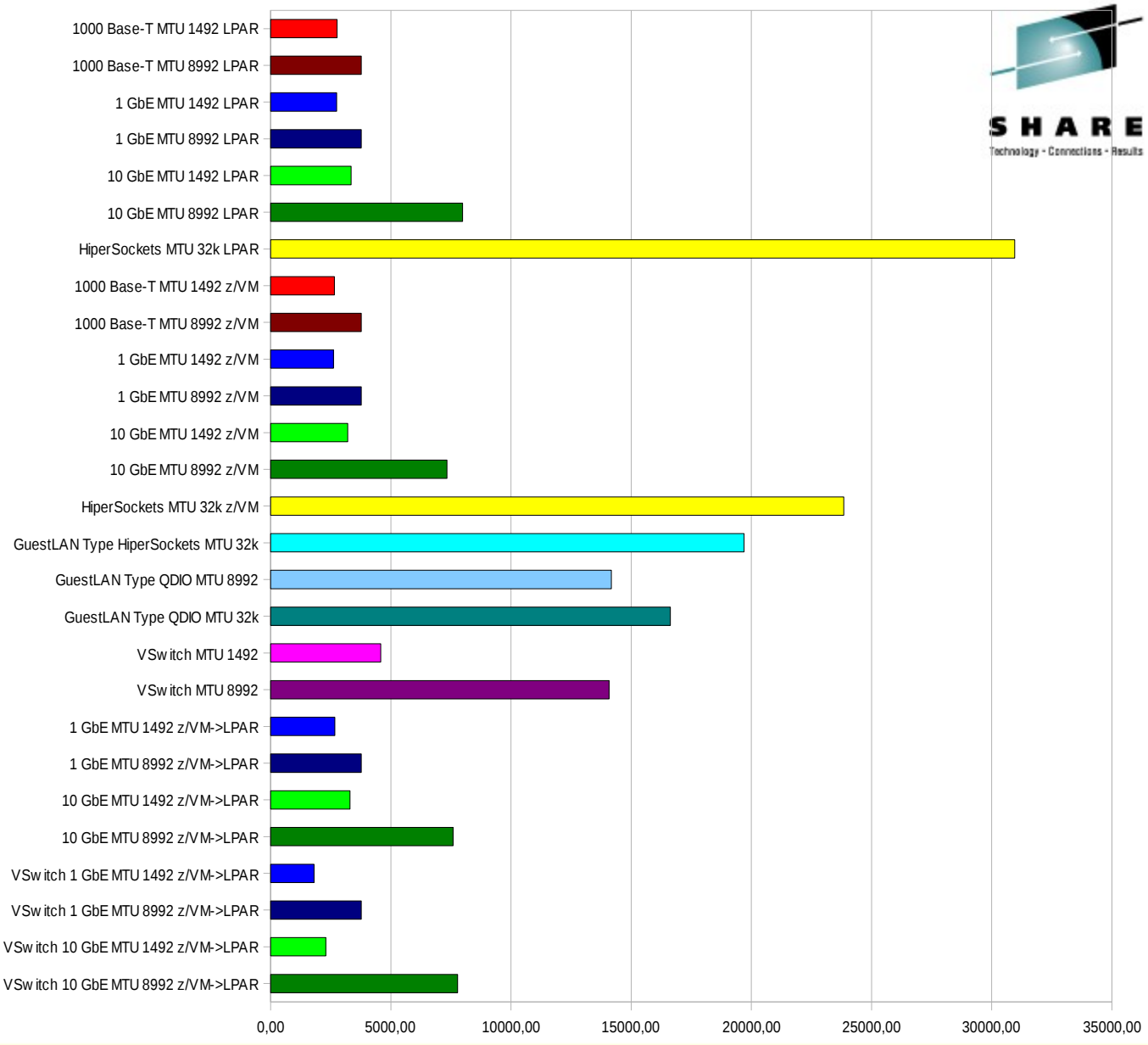
32k response

10 connections

x axis is #transactions

longer bar -> better

- Use large MTU size
- 10 GbE is available
- Use HiperSockets between LPARs
- Use VSWITCH inside z/VM
- Use direct OSA for outside connection of demanding guests



Database query

SLES10 SP2 / z10

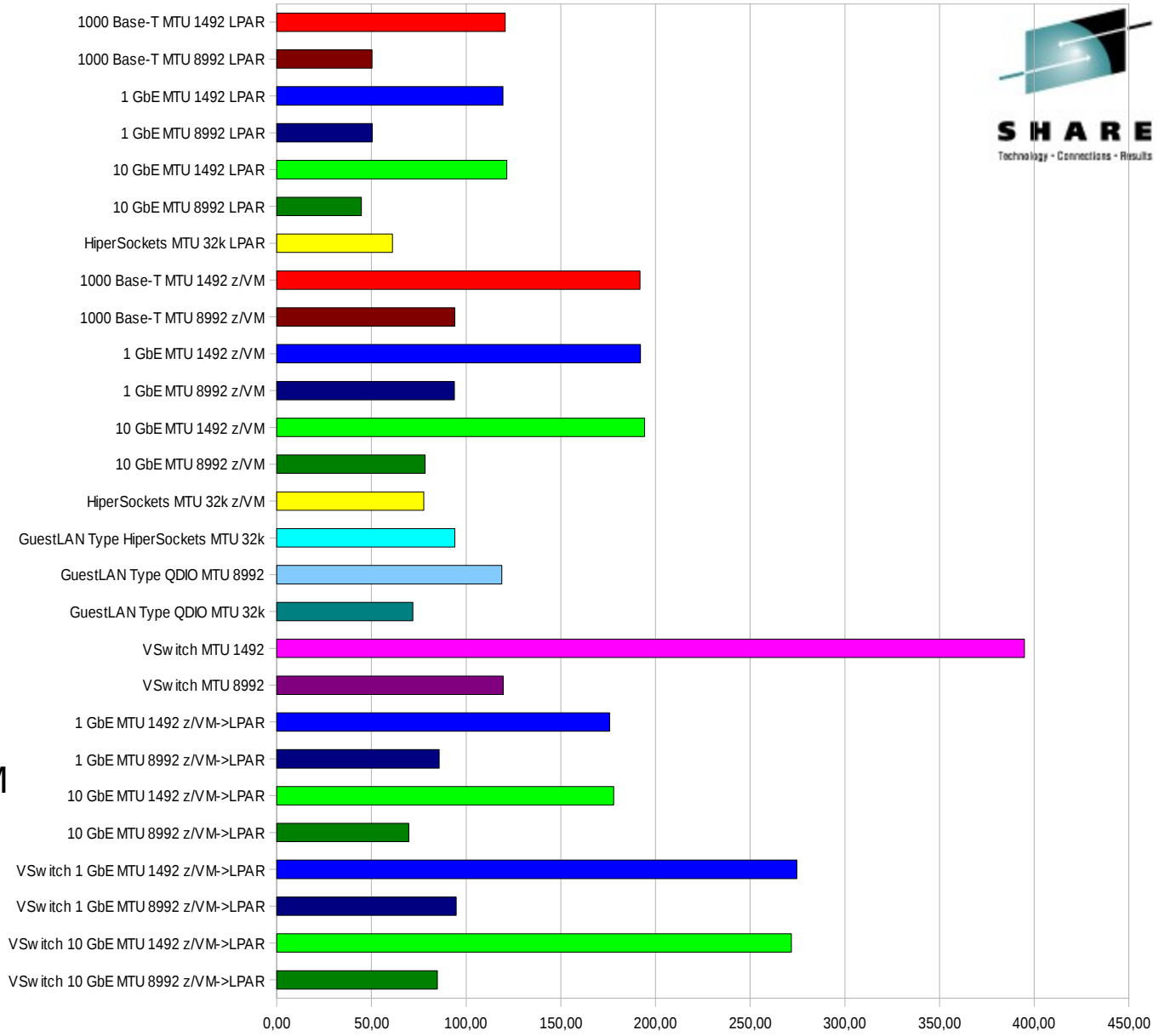
200 byte request

32k response

10 connections

x axis is server CPU utilization per transaction

shorter bar -> better



- Use large MTU size
- 10 GbE is there
- VSWITCH inside z/VM
- VSWITCH for outgoing connections has its price

Online transaction

SLES10 SP2 / z10

200 byte request

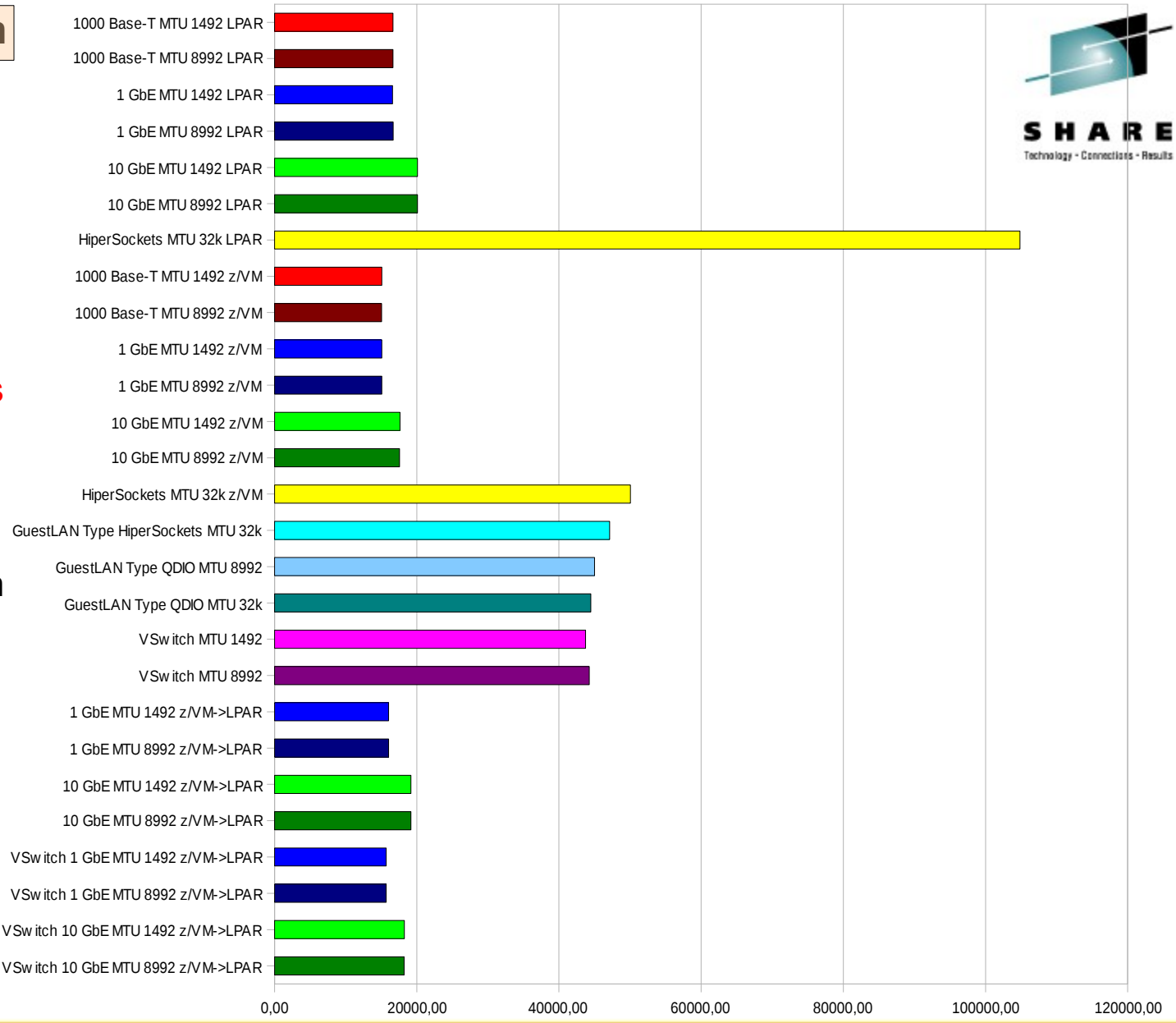
1000 byte response

10 connections

x axis is #transactions

longer bar -> better

- 10 GbE better than 1 GbE
- HiperSockets between LPARs
- VSWITCH inside z/VM
- VSWITCH little bit slower for outside connections



Online transaction

SLES10 SP2 / z10

200 byte request

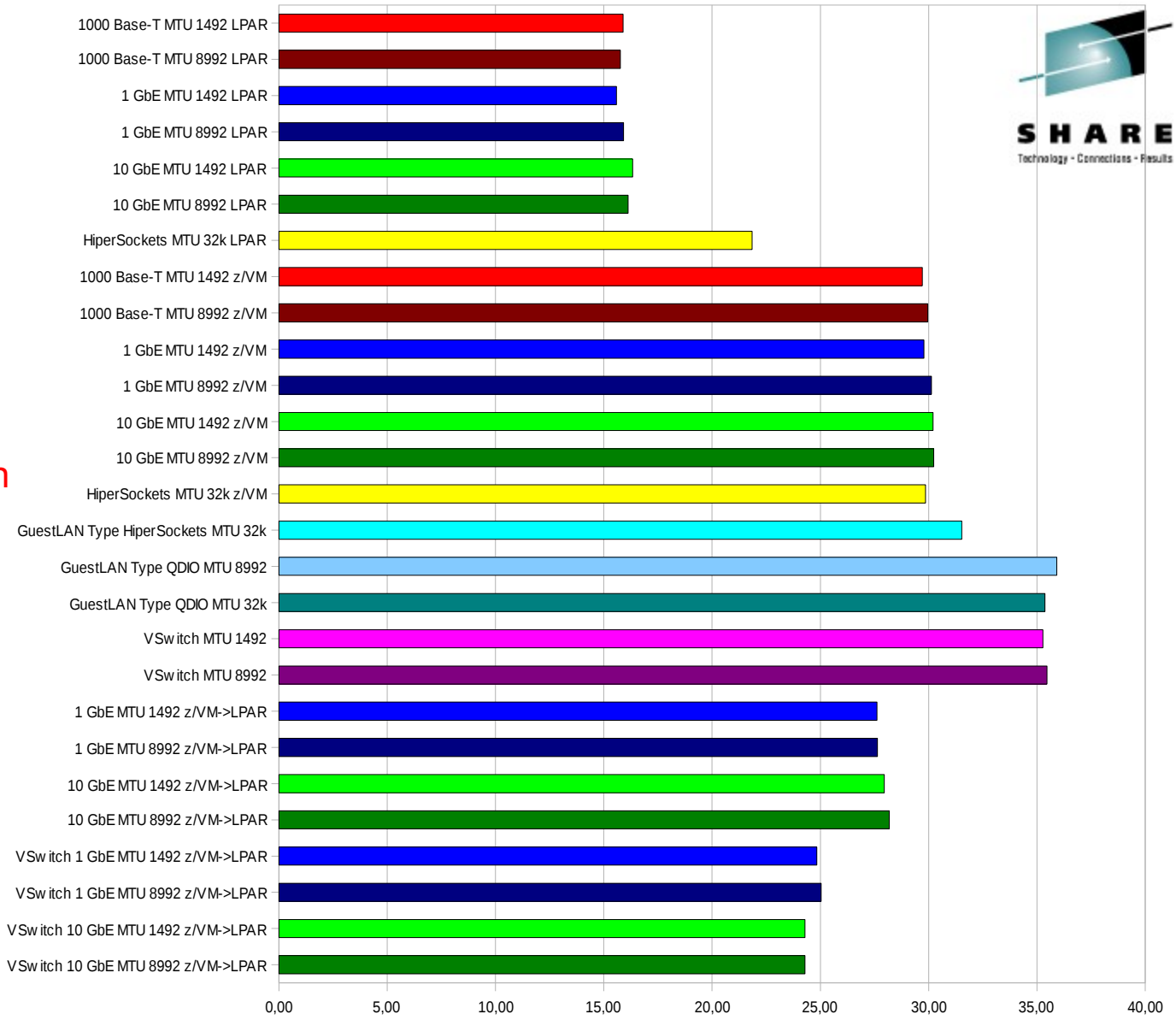
1000 byte response

10 connections

x axis is server CPU utilization per transaction

shorter bar -> better

- Internal connections are expensive
- MTU size doesn't make a difference
- VSWITCH best z/VM option for outside connection



Networking throughput overview (SLES10)

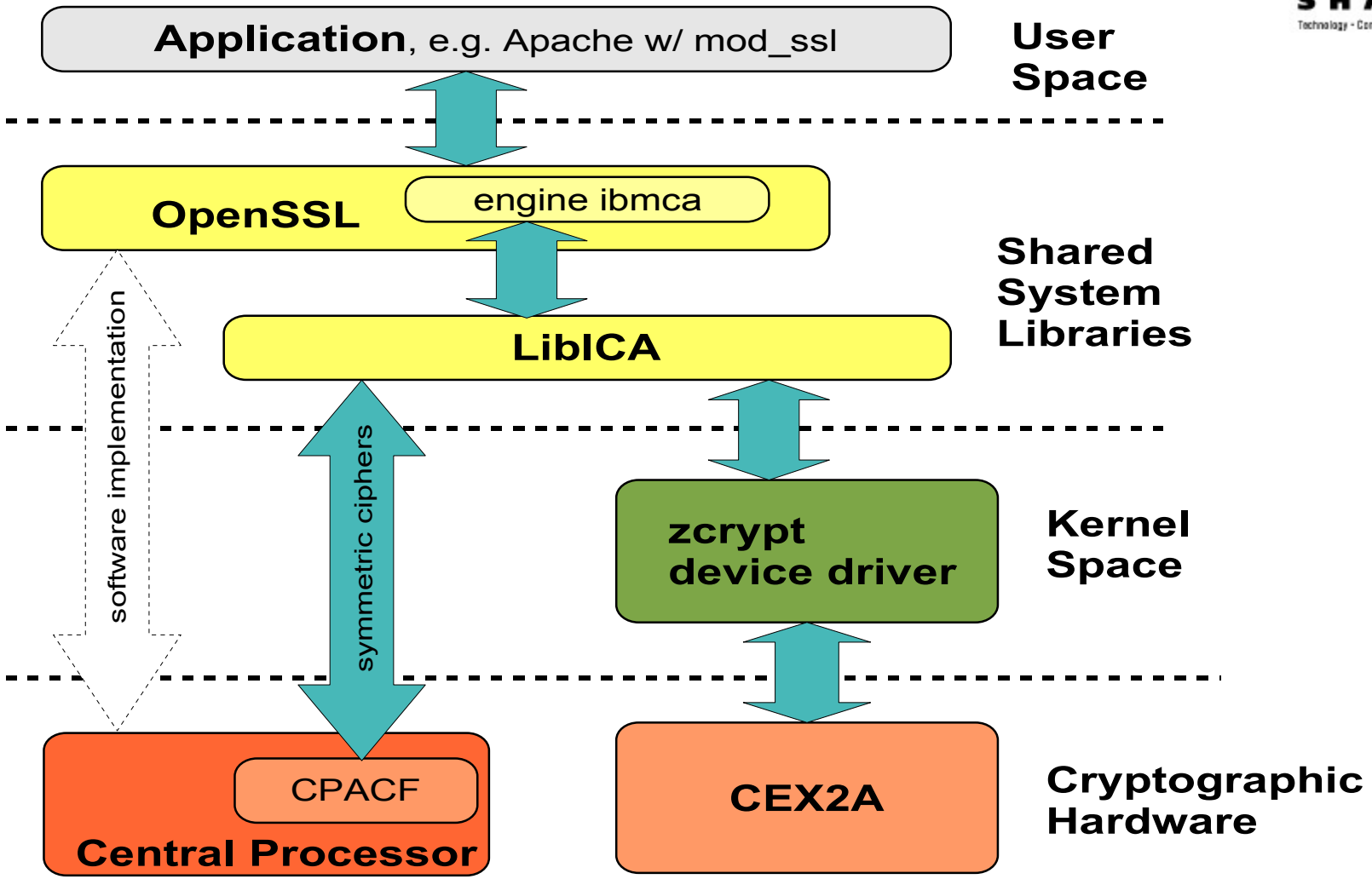
	Website access (crr64x8k)	Online transaction (rr200x1000)	Database query (rr200x32k)	File transfer (strp, strg 20Mx20)
Advantage of large MTU size over default MTU size	1.5x	equal	1.4x (1GbE), 2.3x (10GbE)	3.2x (only 10GbE)
Advantage of 10GbE over 1GbE	1.2x	1.2x	2.1x (large MTU)	3.4x (large MTU)
Advantage of virtual networks over OSA	2.5x (1GbE) 2.0x (10GbE)	2.8x (1GbE) 2.2x (10GbE)	3.7x (1GbE), 1.8x (10GbE)	4.8x (1GbE), 1.4x (10GbE)
Fastest connection	HiperSockets LPAR	HiperSockets LPAR	HiperSockets LPAR	HiperSockets LPAR

- Please see the colorcoding on the previous pages

Agenda

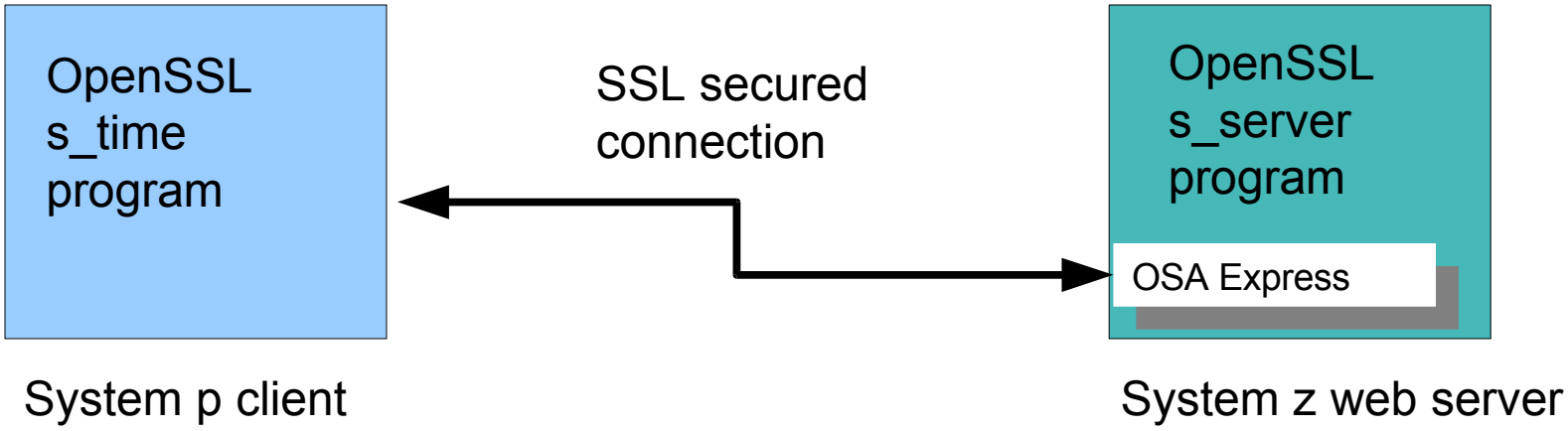
- Network
- **Cryptographic hardware support**

Cryptographic support – Linux SSL stack flow



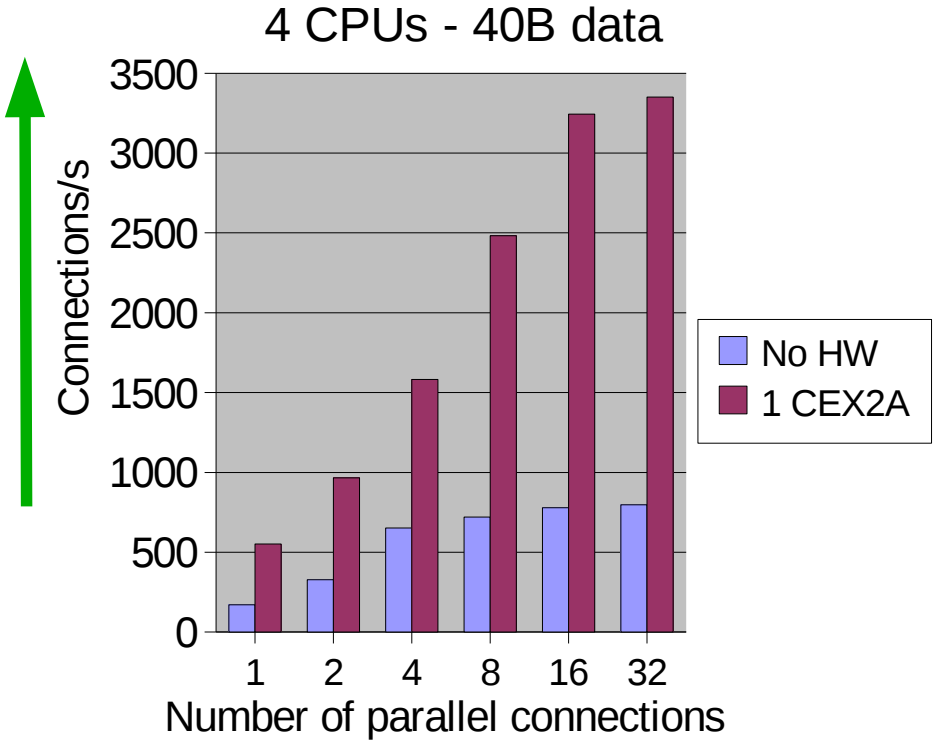
Cryptographic support – Exemplary Workload

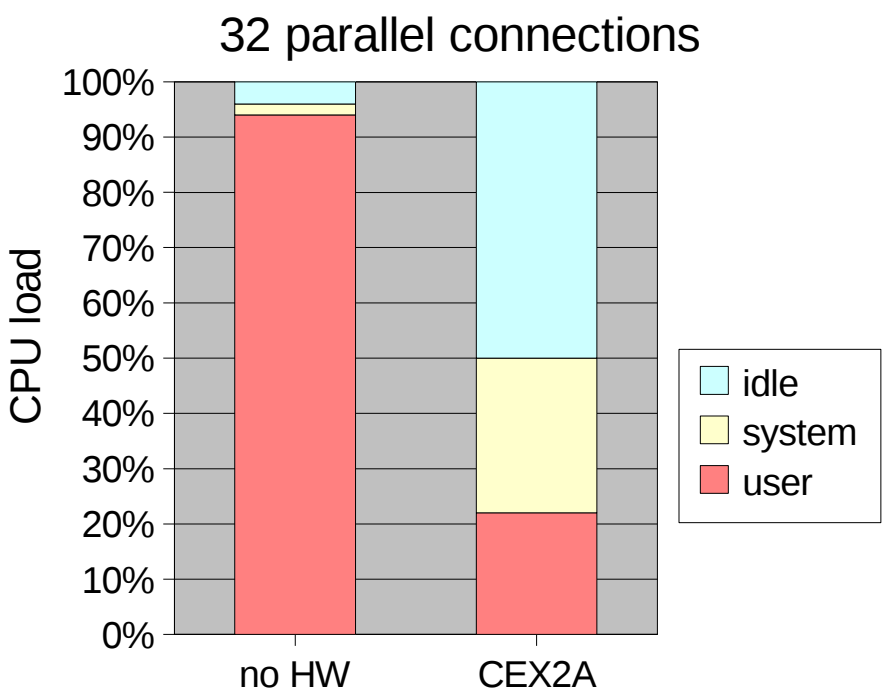
- Workload emulates a secured web server
- Connection between client and server is SSL secured
- Scaling over number of parallel connections
- HTML files of different sizes are exchanged
  - 40 Bytes (SSL handshake)
  - 20 KB (small data portion)
  - 250 KB and 500KB (medium data portions)
  - 1 MB (big data portion)



Crypto Express2 Accelerator (CEX2A) - SSL handshakes

- CEX2A accelerates SSL handshake process (asymmetric cipher RSA)
- Number of SSL handshakes is up to 4x higher with CEX2A support
- In the 32 connections case we save about 50% of the CPU resources



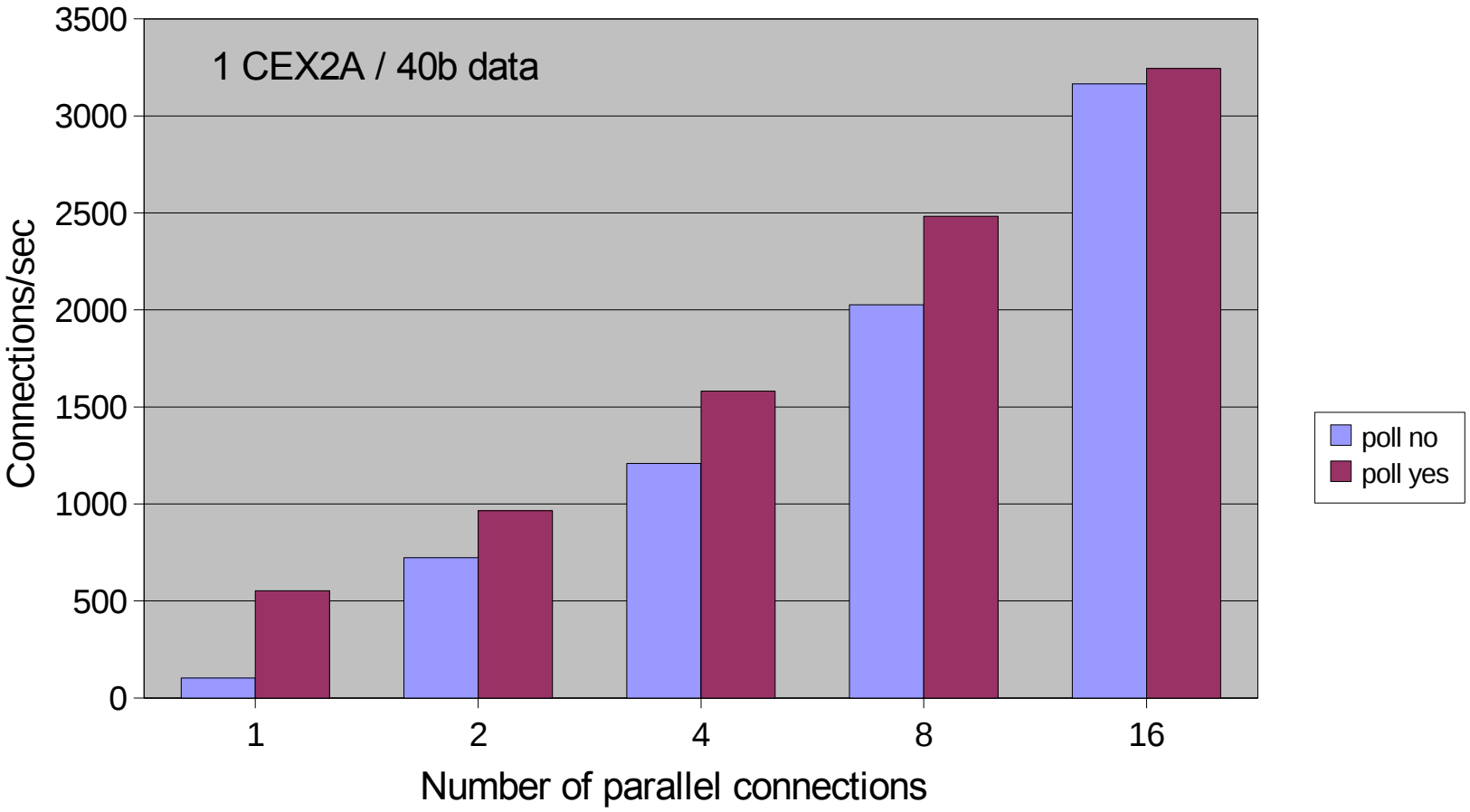


Generic cryptographic device driver polling thread (1)

- zcrypt device driver has a configurable polling thread
  - Introduced with driver version 2.1.0 (SLES10 SP1 and RHEL 5.1);
    - Default was enabled
    - Since SLES10 SP2 and RHEL 5.2 disabled per default
    - **Check state:** `cat /sys/bus/ap/poll_thread` 1==enabled; 0==disabled
- Enabled
  - Polls cryptographic adapter for finished cryptographic requests
  - Best utilization of CEX2 cryptographic adapter
  - Uses one CPU for the thread when polling
  - Only active during outstanding adapter requests
    - **Enable:** `echo 1 > /sys/bus/ap/poll_thread`
- Disabled
  - Finished requests are fetched with Linux timer interrupt
  - Poor performance when cryptographic adapter is not fully utilized
  - No further CPU costs for polling
    - **Disable:** `echo 0 > /sys/bus/ap/poll_thread`

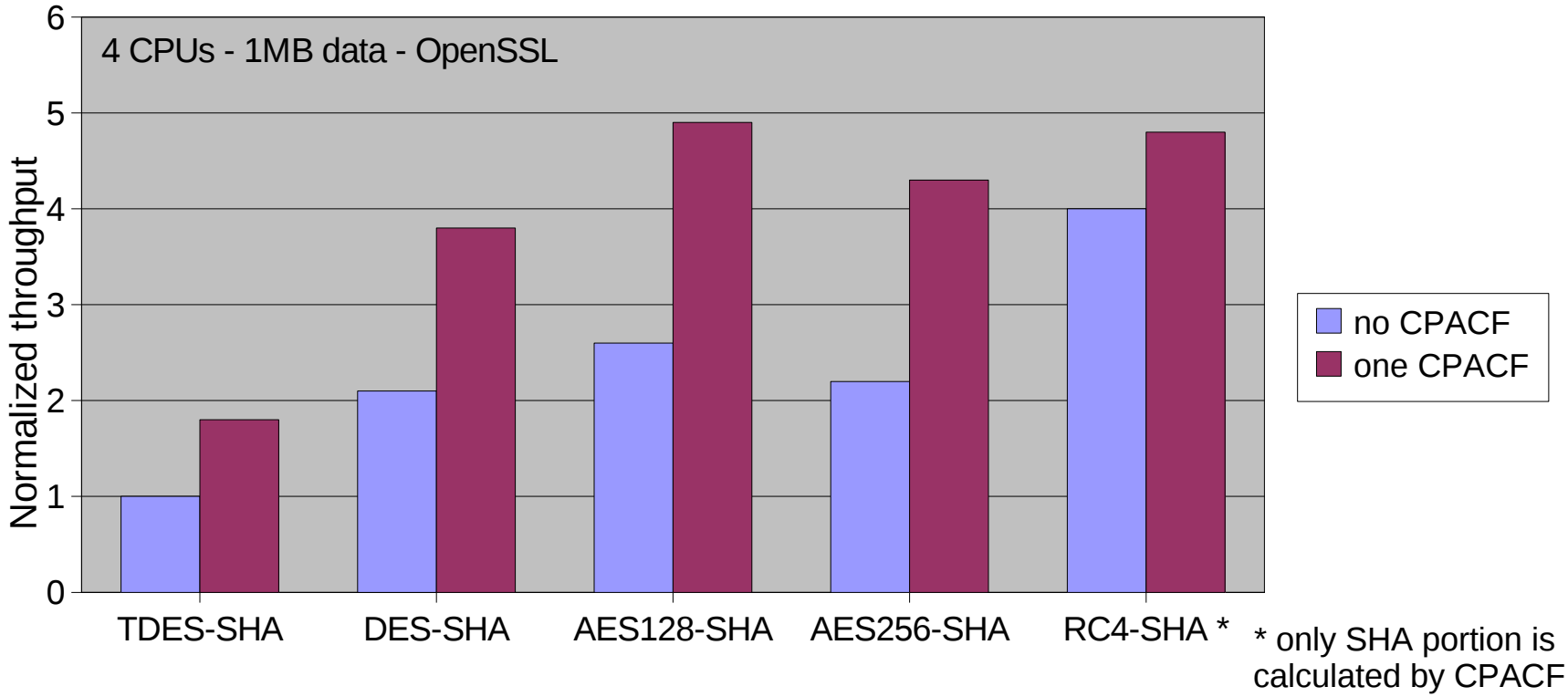
Generic cryptographic device driver polling thread (2)

- Performance degradations until adapter limit is reached for poll=no



CP Assist for Cryptographic Function (CPACF) (1)

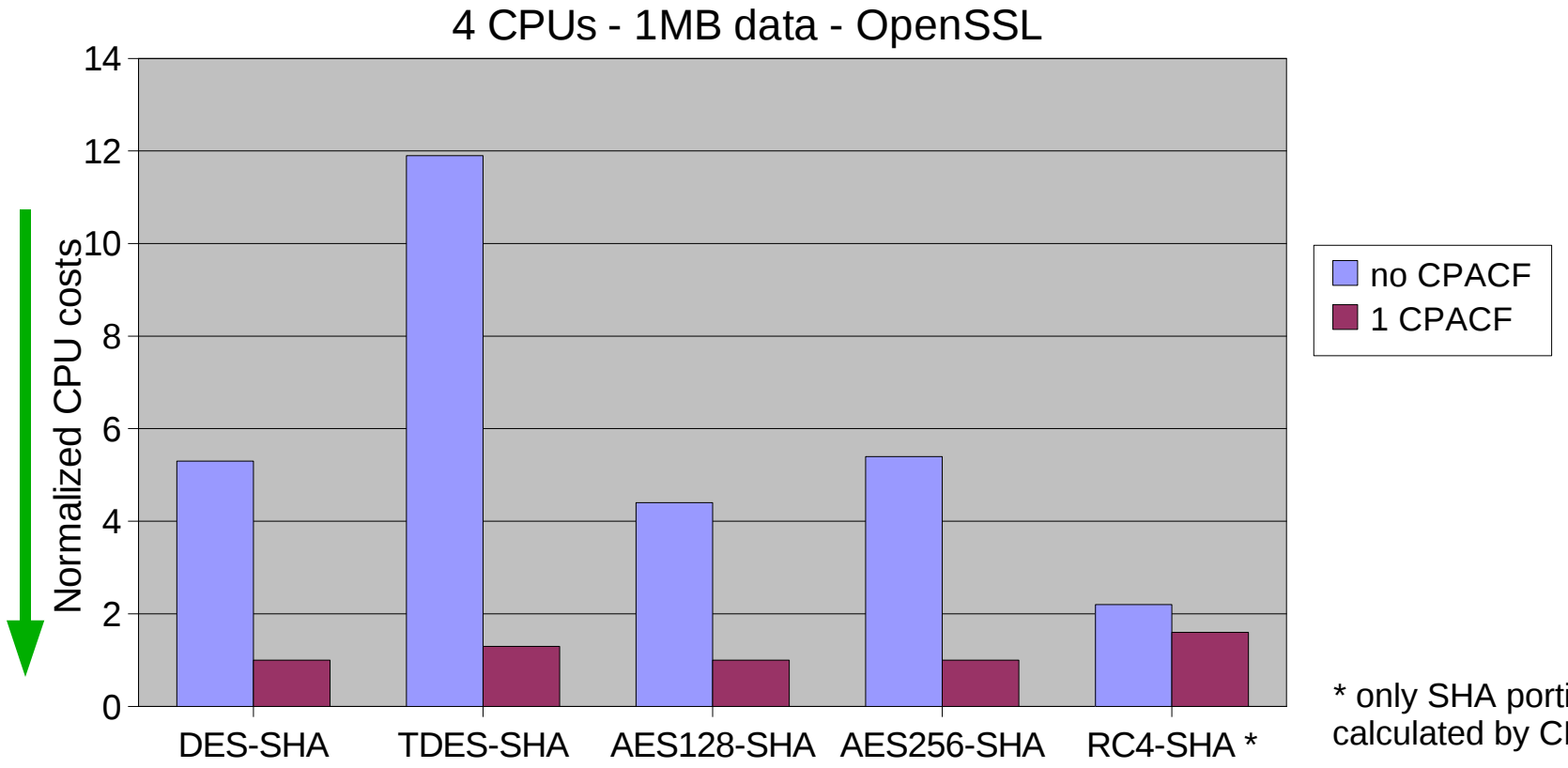
- System z10 supports AES-192, AES-256
- Assure you really use your cryptographic hardware!
  - software configuration issue



* only SHA portion is calculated by CPACF

CP Assist for Cryptographic Function (CPACF) (2)

- Reduced CPU costs for fully supported block ciphers
- TDES most expensive cipher when calculated in software



* only SHA portion is calculated by CPACF

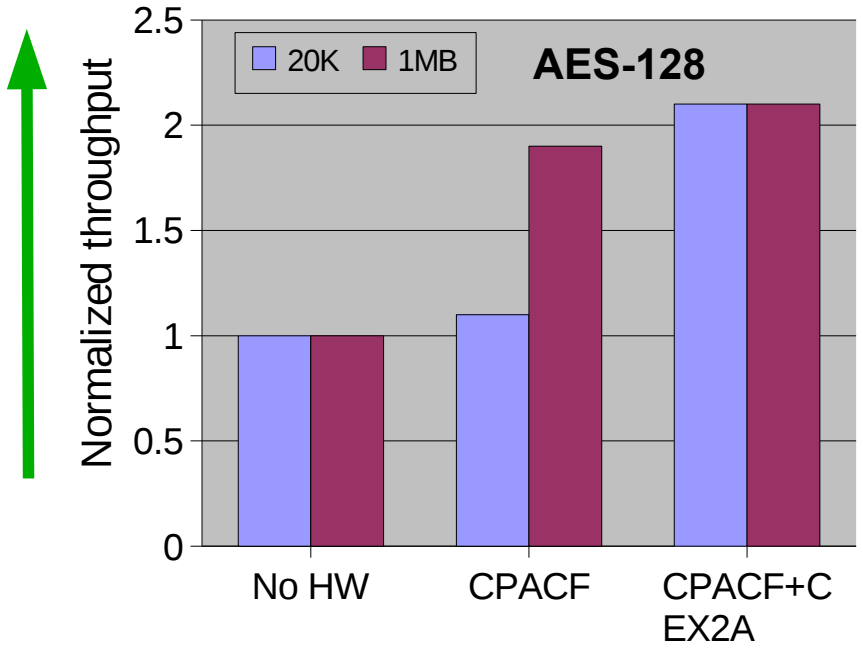
CP Assist for Cryptographic Function (CPACF) (3)

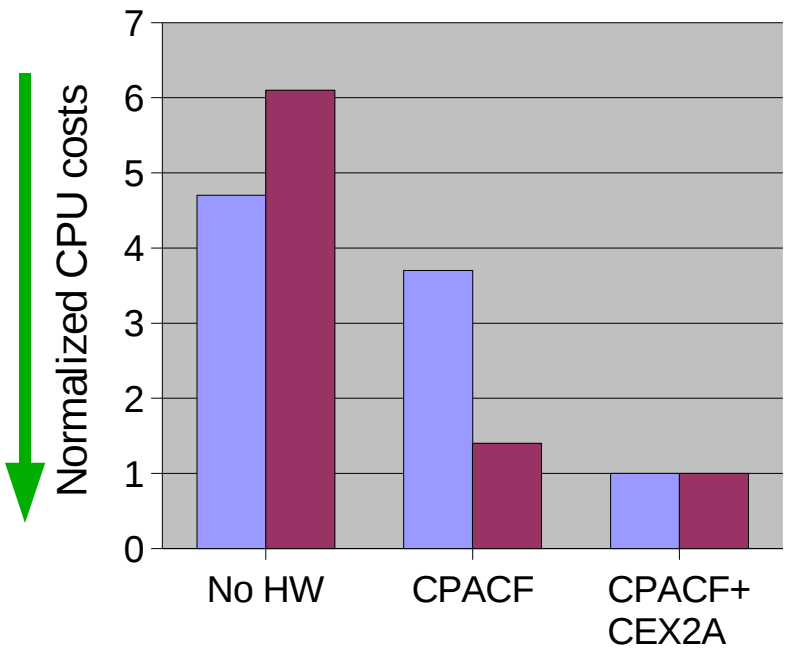
- Supported ciphers and secure hash functions per System z machine

System z machine	supports
zSeries z890, z990	DES, TDES, SHA-1
z9	DES, TDES, AES-128, SHA-1, SHA-256
z10	AES-192, AES-256, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512

SSL traffic with CEX2A and CPACF

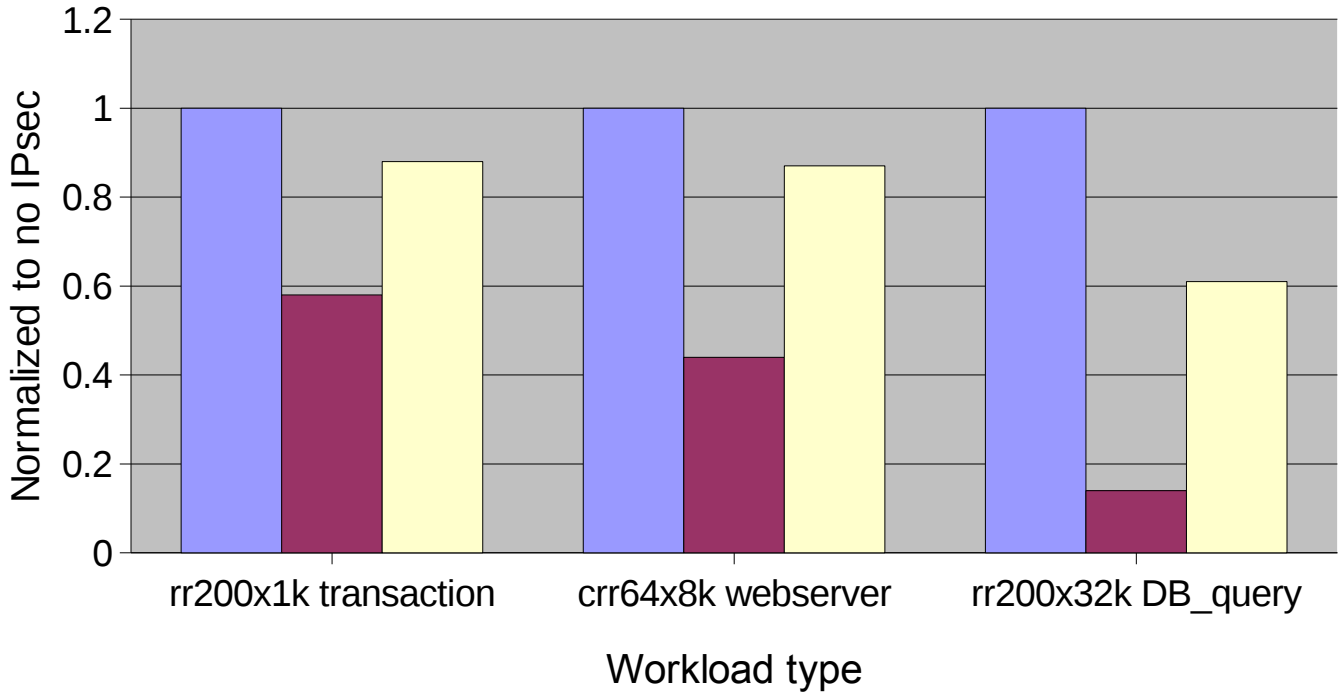
- CEX2A accelerates SSL handshakes
- CPACF accelerates data encryption (symmetric ciphers)
- Use of both hardware features can double the throughput
- Using pure software encryption costs up to 6 x more CPU





Linux in-kernel crypto (1)

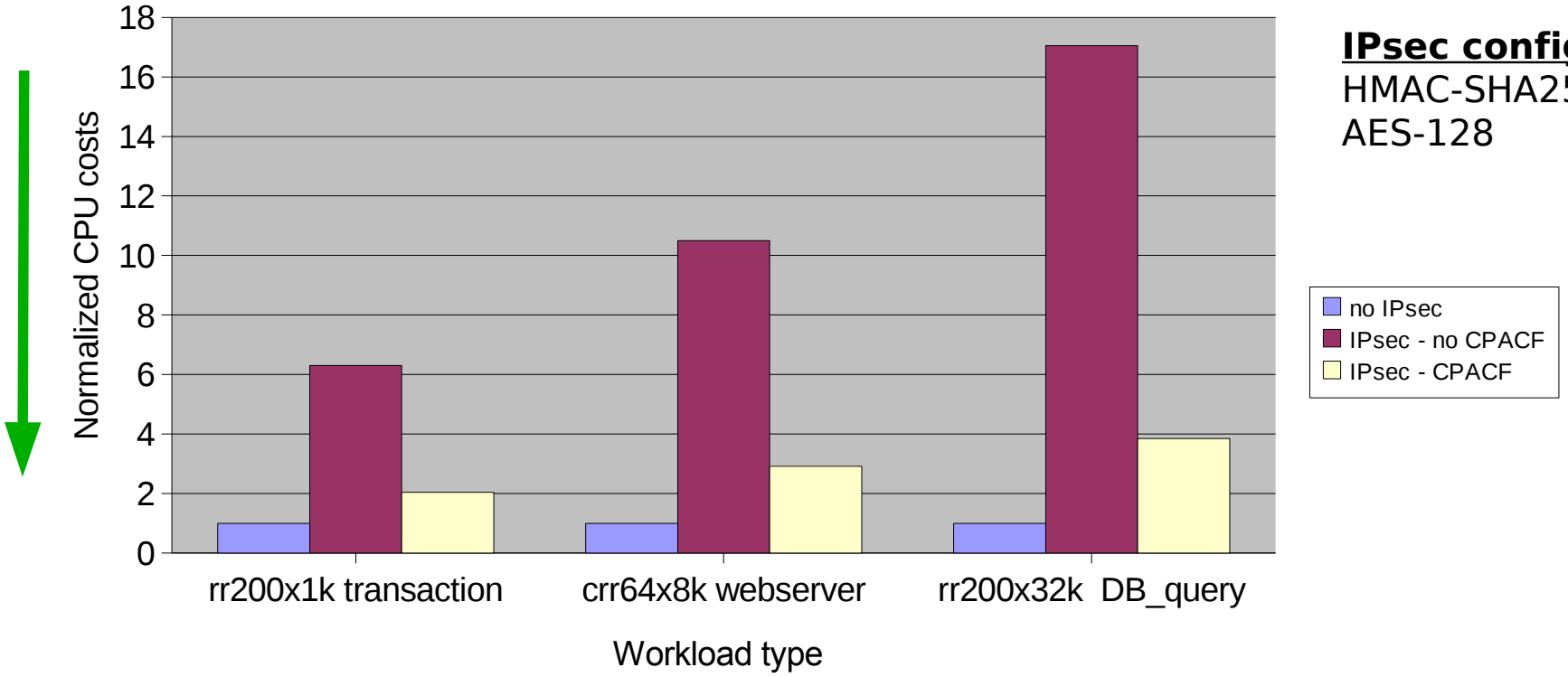
- Example: IP security (IPsec) is done in the Linux kernel
- Linux kernel itself is capable of exploiting CPACF
- Carefully choose a CPACF supported cipher and hash function
- Strong performance impacts with pure software in-kernel crypto



IPsec config:
HMAC-SHA256
AES-128

Linux in-kernel crypto (2)

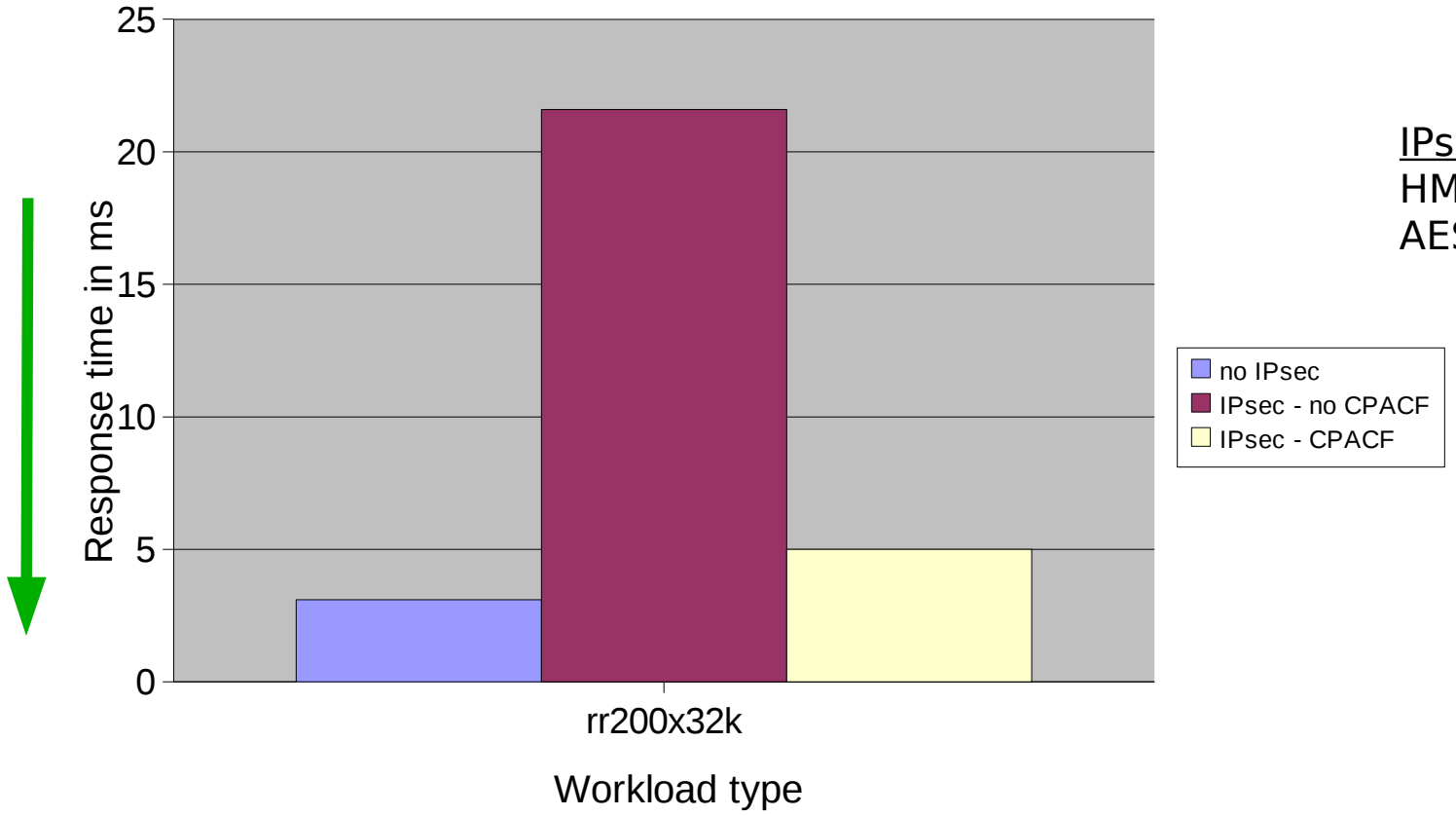
- IPsec overhead is significant
- CPU costs can be 16x higher with software in-kernel crypto
- By using CPACF the CPU costs can be reduced up to 4x



IPsec config:
HMAC-SHA256
AES-128

Linux in-kernel crypto (3)

- Response times <= 5 ms for emulated DB request with IPsec/CPACF



IPsec config:
HMAC-SHA256
AES-128

Linux cryptographic support - Summary

- Crypto Express2 Accelerator (CEX2A)
  - optional feature for System z machines
  - executes cryptographic requests asynchronously to the Central Processor (CP)
  - accelerates public key operations used for the SSL protocol (SSL handshake)
  - requires generic zcrypt device driver
  - zcrypt device driver with enabled polling thread utilizes adapter best
- CP Assist for Cryptographic Function (CPACF)
  - supports several block ciphers and secure hash functions
  - executes cryptographic requests synchronously to the Central Processor (CP)
  - Linux kernel can use CPACF as well (in-kernel crypto)
  - software must be configured appropriately to exploit the hardware
- CEX2A and CPACF can be combined
  - for example: SSL uses symmetric and asymmetric ciphers
  - best throughput results with both cryptographic features together

Visit us !

- Linux on System z: Tuning Hints & Tips
  http://www.ibm.com/developerworks/linux/linux390/perf/
- Linux z/VM Performance Website:
  http://www.vm.ibm.com/perf/tips/linuxper.html
- IBM Redbooks
  http://www.redbooks.ibm.com/
- IBM Techdocs
  http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/Techdocs

Questions