VSE/ESA 2.6 and 2.7

Performance Considerations

Ingo Franzki
e-mail: ifranzki@de.ibm.com
VSE/ESA Development

Trademarks

Contents

- VSE/ESA 2.6
  - Performance Items (Overview)
  - Hardware Support
  - Virtual Tape Support
  - e-Business Enhancements
- VSE/ESA 2.7
  - Performance Items (Overview)
  - HiperSockets
  - Hardware Crypto support
- z800/z900 Remarks
- Dependencies for VSE/ESA Growth

VSE/ESA 2.6 Performance Items

- VSE/ESA 2.6 Base enhancements
  - Delete Label Function
  - LTA Offload for some AR commands
  - SVA-24 Phases moved above the line
  - Increased max number of SDL entries
  - SDL update from non-BG partitions
  - POWER Data file extension without reformat

VSE/ESA 2.6 Performance Items - continued

- VSE/ESA 2.6 Hardware Support
  - FICON Support (VSE/ESA 2.3 or higher)
  - New 2074 System Management Console
  - OSA Express Adapter (e.g. Gigabit Ethernet)
  - VSAM Support for large 3390-9 Disks (Shark)
  - Fastcopy Exploitation of ESS FlashCopy and RVA SnapShot

VSE/ESA 2.6 Performance Items - continued

- VSE/ESA 2.6 e-Business Enhancements
  - Updated Java-based connectors
  - VSAM SHROPT(4) avoidance for connectors
  - SSL for VSE/ESA exploitation
  - SSL enabled CICS Web Support
  - CICS External Call Interface
  - New VSAM Redirector
  - More samples (JConVSE, VSEPrint, etc.)
  - New JDBC Driver Layer for VSAM

Delete Label Function

- New function DELLBL in LABEL macro
- Must be explicitly exploited
  - Important for vendors with disk/tape management products
- Benefits
  - Saves recursive reads (GETNXGL) and write backs (ADDLBL/ADDNXL)
  - Saves >90% of the SVCs for this activity
- More Info
  - VSE Label Area -Layout and Capacity Consideration, VSE/ESA Software Newsletter, 12/2000
  - http://www-1.ibm.com/servers/eserver/zseries/os/vse/pdf/vsenew21/vseflab.pdf

LTA Offload and SVA-24

- LTA Offload for some AR commands
  - Phases $$BATTNC and $$BATTNG are merged into $$BATTNA
  - Code of $$BATTNB is merged into IJBAR
  - Benefits
    - Less I/O by less FETCHes for LTA load
      - IGNORE, PAUSE, LOG, NOLOG, NEWVOL, START, BATCH
    - No LTA usage for MSG commands
- SVA-24 Phases moved above the line
  - $IJBPRTY (6K)

SDL Entries

- Increased max. number of SDL entries
  - New IPL SVA parameter: SDL=n
    - Maximum value now 32765
    - About 56 SDL entries per 4K page in shared space below
    - Theoretically would cost 2.28 MB
- SDL updates from non-BG partitions
  - SET SDL command can now be issued from any partition
  - Internal locking is done to assure correctness

Hardware Support

- New 2074 System Management Console
  - ESCON channel attached
  - Eliminates requirement for a non-SNA 3174 controller
- OSA Express Adapter Support
  - Available for G5 and above

	Gigabit Ethernet	Fast Ethernet 100 Mbps	ATM-LE 155 Mbps	Tokenring 4/16/100 Mbps
CHIPID TYPE=OSE (non-QDIO)	no	yes	yes	yes
CHPID TYPE=OSD (QDIO)	yes	yes	yes	yes

Hardware Support

- Queued Direct I/O
  - Designed for very efficient exchange of data
  - Uses the QDIO Hardware Facility, without traditional S/390 I/O instructions
  - Without interrupts (in general)
  - Use of internal queues
  - With pre-defined buffers in memory for asynchronous use
- Exploitation by TCP/IP for VSE/ESA
  - see TCP/IP Performance Considerations

ESS Flashcopy

- The DASD Architecture of ESS allow copy of DASD's with the utility FlashCopy
  - The copy process takes a few seconds instead of hours
  - From Operating system view it is a real copy
  - From DASD controller view it is a virtual copy
- FlashCopy support is available for 3 VSE products
  - IXFP SNAP command
  - VSAM SYNONYM Backup
  - VSE/Fast Copy

ESS Flashcopy - continued

- Problems
  - Duplicate VOLIDs (DASD names) not allowed on a VSE system
  - Duplicate VSAM Catalog names not allowed on a VSE system
- FlashCopy of volume containing VSAM datasets would mean
  - duplicate VOLIDs
  - duplicate VSAM Catalog names
- Solution
  - IDCAMS SNAP command
    - Changes the VOLIDs of the copied volumes
  - IDCAMS SYNONYM BACKUP command
    - Uses a synonym list to access copied volumes

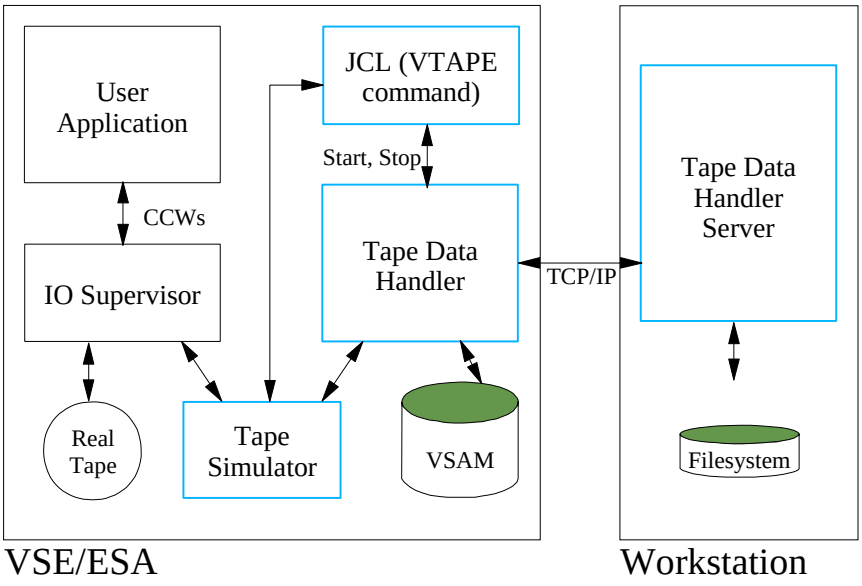
Virtual Tape Support

- Allows access to tape images residing in
  - A VSAM file (ESDS)
  - A remote file on a workstation or server
- Tape image is AWSTAPE format
  - Known from P/390, R/390 or FLEX/ES
- New VTAPE command
- Virtual Tape Simulator
  - Simulates channel program execution
- Virtual Tape Data Handler
  - Runs in a partition

Virtual Tape Support - continued

- Virtual Tape Server
  - Runs on a workstation or server (Java)
  - Allows to access a tape image remotely
  - Communicates via TCP/IP with Virtual Tape Data Handler
- Designed to allow e-Delivery and e-Service (future)
  - Download a tape image containing a product
  - Obtain a CD/DVD containing the tape image
  - Install the product via Virtual Tape directly from the workstation
- Also possible
  - Backup to a Virtual Tape + copy to CD
  - Restore directly from CD via Virtual Tape

Virtual Tape Support - continued



Updated Java-based Connector

- The Java-based connector has been updated to support the Java 2 platform (JDK 1.3)
- Introduced JDBC layer for VSAM access
  - Allows to issue SQL statements
- Adaptations for WebSphere 4.0
  - Enhanced connection pooling by support of JCA (Java Connector Architecture)
  - Connectors can be deployed as Resource Adapter and as (JDBC-) Data Source
- SSL enabled connections possible
  - Transparent use of secured connections

VSAM Share Options with Connectors

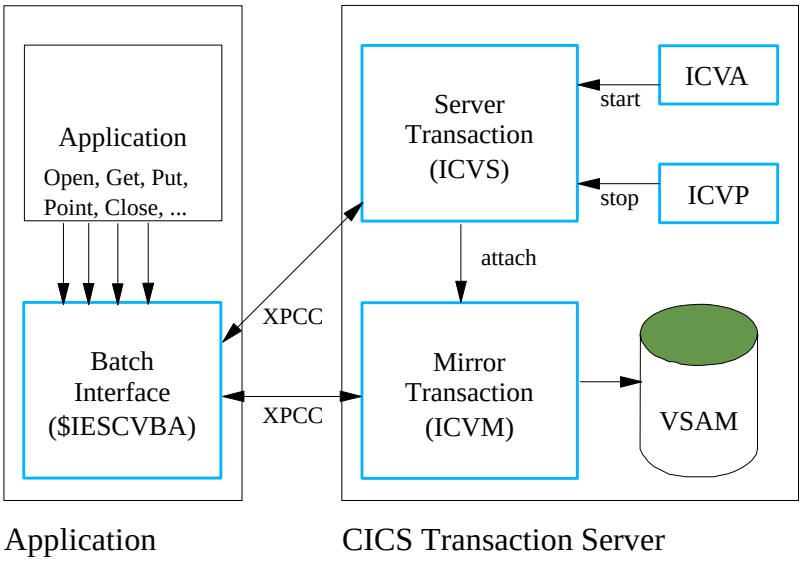
- SHROPT(4) Backgrounds
  - Using connectors to UPDATE a VSAM file already opened for output (e.g. by CICS) needs SHROPT(4)
  - SHROPT(4) has big overhead
- Performance implications
  - Bigger pathlength for processing of UPDATE requests due to VSAM internal locking
  - Each READ must be done from disk
  - Each WRITE must go to disk
  - Additional catalog I/Os for statistics
  - Influence on any application, not only connectors

VSAM SHROPT(4) Avoidance

- Connectors in VSE/ESA 2.5 require SHROPT(4) when updating VSAM files owned by CICS
- New VSAM-via-CICS Service avoids SHROPT(4) by routing the VSAM requests to CICS
- Communication between batch and CICS is XPCC
- New transactions related to VSAM-via-CICS:

Transaction	Program	Description
ICVA	IESCVSTA	starts the service
ICVP	IESCVSTP	stops the service
ICVS	IESCVSRV	internal server task
ICVM	IESCVMIR	internal mirror task
none	IESCVSTI	internal start program

VSAM-via-CICS Service



VSAM-via-CICS Service - continued

- How VSAM-via-CICS works
  - Long running server transaction ICVS
  - Attaches a mirror transaction ICVM on request
  - Mirror transaction is attached for
    - "Open" from batch
    - Browse files from batch
  - Mirror transaction ends at "close" from batch
  - Service can run in multiple CICSes at the same time
  - Batch counterpart is implemented in phase $IESCVBA

VSAM-via-CICS Service - continued

- Naming convention for "VSAM-via-CICS files"
  - Each CICS is treated as "virtual" catalog
  - Files defined in CICS (via CEDA DEFINE FILE) are visible within this catalog
  - "Virtual" catalog file id

    #VSAM.#CICS.<applid>

    indicates "virtual" CICS catalog

    APPLID of CICS region owning the files within this catalog
- "Virtual" cluster file id is the 7 character name known in CICS

VSAM-via-CICS Service - continued

- Example
  - Assume there is a CICS region DBDCCICS
  - CICS knows a file named MYFILE
  - Real VSAM files MY.VSAM.TEST.FILE resides in catalog MY.USER.CATALOG
  - "Batch only" name would be
    - Catalog: MY.USER.CATALOG
    - Cluster: MY.VSAM.TEST.FILE
  - "VSAM-via-CICS" name would be
    - Catalog: #VSAM.#CICS.DBDCCICS
    - Cluster: MYFILE

VSAM-via-CICS Service - continued

- VSAM-via-CICS files can only be accessed from the following applications
  - Java-based connector via VSE Java Beans
  - DB2-based connector via VSAM CLI (SQL)
  - REXX new VSAMIO function
- IDCAMS does NOT show these files
- "Virtual" names can NOT be specified in DLBLs
- No changes made in VSAM for this support
- No influence on "normal" VSAM processing
- But: Maps can be defined for a "virtual" file
  - Via Java-Based connectors
  - Via IDCAMS RECMAP function

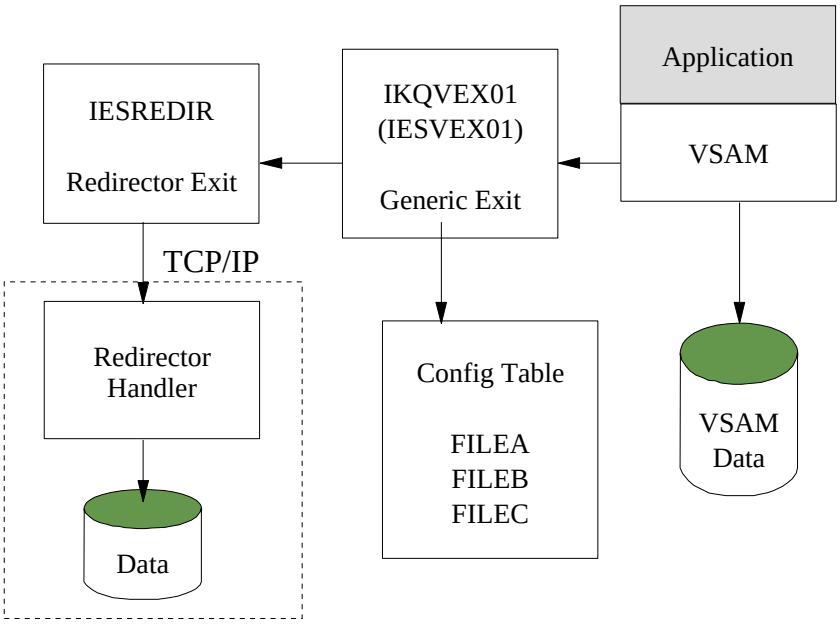
VSAM Redirector

- New connector
  - VSE is client
  - PC / workstation is server
- Exploits VSAM exit IKQVEX01
- Allows to redirect one of more VSAM files to a PC or workstation
- All VSAM requests of a particular file are redirected
  - Open / close
  - Get / put / point / delete / insert
- Transparent for applications
  - Usable from batch and CICS

VSAM Redirector - continued

- Owner of data can be
  - VSAM
    - Requests are forwarded to workstation
    - VSAM still owns the data
    - VSAM executes the requests
    - Used for data replication/syncronisation
  - PC / workstations
    - VSAM does not execute the requests
    - Handler on workstation 'simulates' VSAM logic
    - A VSAM file with at least one dummy record is required (for open processing)

VSAM Redirector - continued



VSAM Redirector - continued

- Decision if a file is redirected or not is
  - Done at open time
  - Based on the config table (PHASE)
    - Catalog id and file id
  - Only a very small (open-)overhead for non-redirected files
  - No overhead for get/put/... if not redirected
- Generic Exit can also call a 'Vendor' exit instead of Redirector exit
  - Defined in the config table
  - Based on catalog id and file id

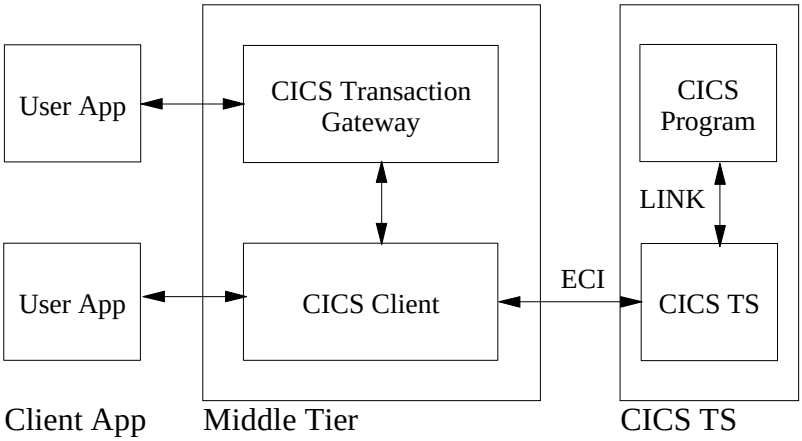
VSAM Redirector - Performance Implications

- Is the file redirected ?
  - ►No: only at OPEN time (very small overhead)
  - ►Yes: at each request
- Network overhead ?
  - ►Yes, if file is redirected
  - ►Depends on
    - –Number of VSAM requests
    - –Size of records
- Data ownership
  - ►OWNER=REDIR
    - –no VSAM I/O

CICS TS Enhancements

- CICS Web Support
  - ►new: SSL enabled (https)
- External Call Interface (ECI)
  - ►Call a CICS program from a workstation
  - ►Prerequisits
    - –CICS Client
    - –CICS Transaction Gateway

External Call Interface (ECI)



General Performance Hints for Connectors

- Reduce amount of data transferred
  - Transfer only data that is needed
  - Issue only requests that are needed
- Use connection pooling
  - Reduce overhead of connection establishment
- Performance of connectors depends on
  - Network performance
  - Performance of "server"
  - Performance of "client" or middle tier

VSE/ESA 2.7 Performance Items

- VSE/ESA 2.7 hardware support
  - z800/z900, Multiprise 3000, G5/G6
  - HiperSockets
  - Hardware Crypto Support
  - 32760 cylinder 3390 support
  - 3590 buffered tape mark
- VSE/ESA 2.7 enhancements
  - New TCP/IP for VSE/ESA release 1.5
  - $IJBLBR above the line
  - II User Status Record above the line
  - VTAPE: removed DVCDN/DVCUP
  - POWER: reallocate queue file during warm start

VSE/ESA 2.7 Hardware support

- VSE/ESA 2.7 runs on the following machines
  - z800 (2066)
  - z900 (2064)
  - 9672 Parallel Enterprise Server (G5/G6)
  - Multiprise 3000 (7060)
  - equivalent emulators (Flex-ES)
- VSE/ESA 2.7 is based on the hardware instruction set described in the manual 'ESA/390 Principles of Operation' (SA22-7201).
- With VSE/ESA 2.7 it is assumed that all the ESA/390 instructions and facilities described in that manual can be used.

z800/z900 Remarks

- Prior to z800/z900 there is one cache for data and instructions
- z800/z900 has split data and instruction cache
- Performance implications:
  - If program variables and code that updates these program variables are in the same cache line (256 byte)
    - Update of program variable invalidates instruction cache
    - Performance decrease if update is done in a loop
  - See APAR PQ66981 for FORTRAN compiler

32760 cylinder 3390 support

- With announcement 101-341 at 11/13/2001 IBM announced the new 32760 cylinder 3390 volumes of the IBM TotalStorage Enterprise Storage Server (ESS)
  - This enhancement of the ESS F models was made available 11/30/2001
- VSE/ESA 2.7 now supports these volumes
  - helps relieve address constraints
  - improves the disk resource utilization
  - can be used to consolidate multiple disk volumes into a single address
- VSAM can only address 10017 cylinders.

3590 Buffered Tape Mark support

- The 3590 control unit provides support for writing tape marks (TM) in buffered mode
- Writing TM's in "buffered" mode should enhance the performance
  - of all programs which write many TM's as part of their file creation process (e.g. POFFLOAD)
- All the TM's written during OPEN/CLOSE (label processing) will remain to be written "UNbuffered"
  - all the programs which write TM's mainly or only during OPEN/CLOSE will NOT benefit from this enhancement

$IJBLBR phase moved above the line

- The $IJBLBR.PHASE has been split into two phases
  - $IJBLBR.PHASE
  - $IJBLB31.PHASE
- $IJBLBR.PHASE will continue to reside in SVA-24
- $IJBLB31.PHASE will reside in SVA-ANY (high SVA)
  - This will free about 180KB in SVA-24

II User status record above the line

- During Logon each II user gets besides others two storage areas allocated
  - User_Status_Record USR (904 bytes)
  - Panel_Hierarchy_List PHL (1352 bytes)
  - originally located in the CICS DSA (below)
- With VSE/ESA 2.7 the USR and PHL has been moved to ESDSA (shared above)
  - frees 2.3 KB in DSA below per user
- ICCF TCTUALOC=ANY now supported
  - ICCF transaction programs has been changed to support a TCTUA (28 bytes) above the line

HiperSockets hardware elements ('Network in a box')

- Synchronous data movement between LPARs and virtual servers within a zSeries server
  - Provides up to 4 "internal LANs" HiperSockets accessible by all LPARs and virtual servers
  - Up to 1024 devices across all 4 HiperSockets
  - Up to 4000 IP addresses
  - Similar to cross-address-space memory move using memory bus
- Extends OSA-Express QDIO support
  - LAN media and IP layer functionality (internal QDIO = iQDIO)
  - Enhanced Signal Adapter (SIGA) instruction
    - No use of System Assist Processor (SAP)

HiperSockets hardware elements ('Network in a box') - continued

- HiperSockets hardware I/O configuration with new CHPID type = IQD
  - Controlled like regular CHPID
  - Each CHPID has configurable Maximum Frame Size
- Works with both standard and IFL CPs
- No physical media constraint, no physical cabling, no priority queuing
- Secure connections

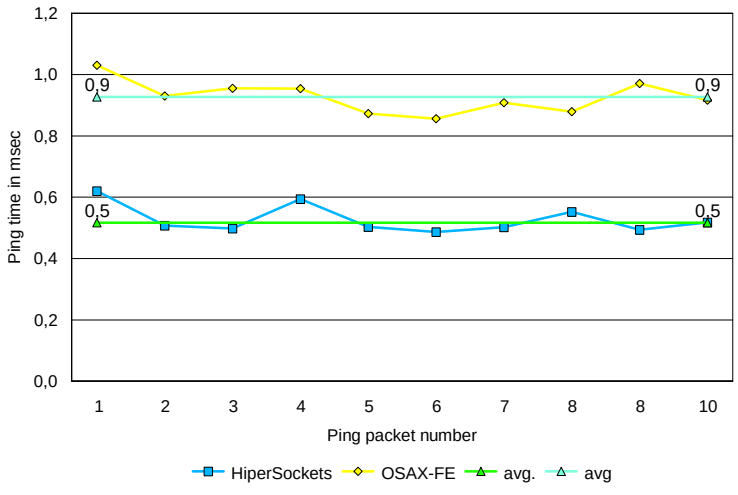
Measurement Environment

- z800 (2066-004)
  - 4 processors
- VSE/ESA 2.7 GA Driver in an LPAR (native)
  - 1 CPU active (~2066-001)
  - TCPIP00 (F7): OSA Express Fast Ethernet
  - TCPIP01 (F8): HiperSockets
- Linux for zSeries in an LPAR (native)
  - 3 CPUs active (shared)
  - eth0: OSA Express Fast Ethernet
  - hsi10: HiperSockets

Latency (Round trip time) - results

- Measurements has been done with PING command
  - Issued at Linux side
  - 10 Pings
  - PING sends a datagram to VSE
  - VSE sends an answer back to Linux
  - Time until answer arrives is measured
    - Round trip time

Latency (Round trip time) - results

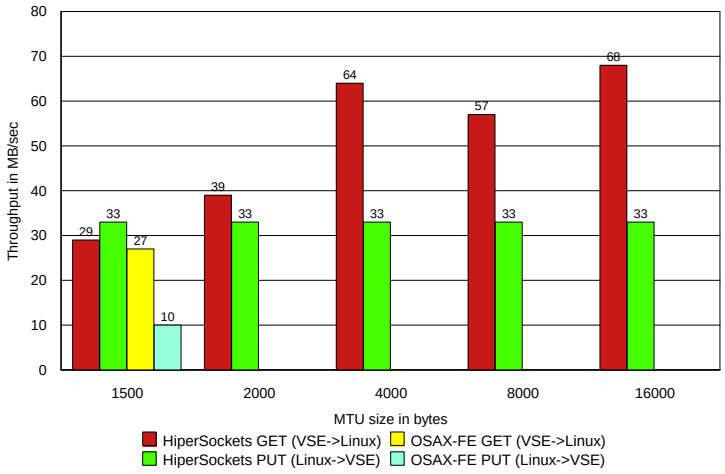


HiperSockets is about 1.8 times faster in terms of latency

Throughput (MB/sec)

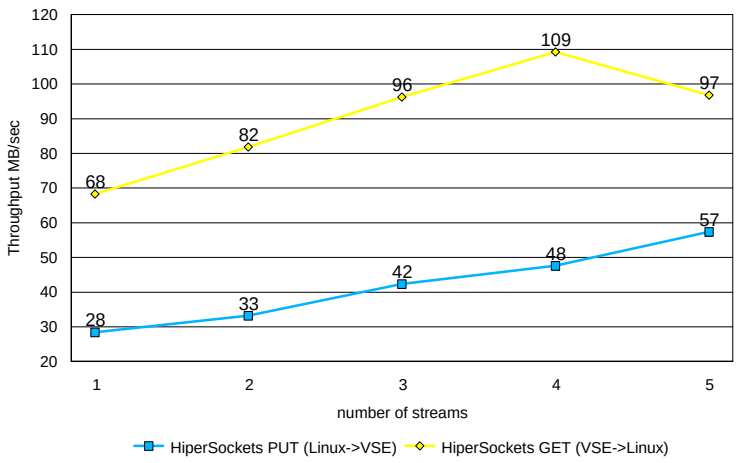
- Measurements has been done with FTP
  - Initiated at the Linux side
  - Transferring 1GB (1000MB)
    - without translation (binary)
    - 1 to 5 parallel streams
  - PUT: send data to VSE
    - VSE inbound
    - sending a 1GB file to $NULL file (in memory file)
    - No file I/O is done by VSE/Linux
  - GET: receive data from VSE
    - VSE outbound
    - receiving $NULL file (in memory file) into /dev/null
    - No file I/O is done by VSE/Linux

Throughput (MB/sec) - results



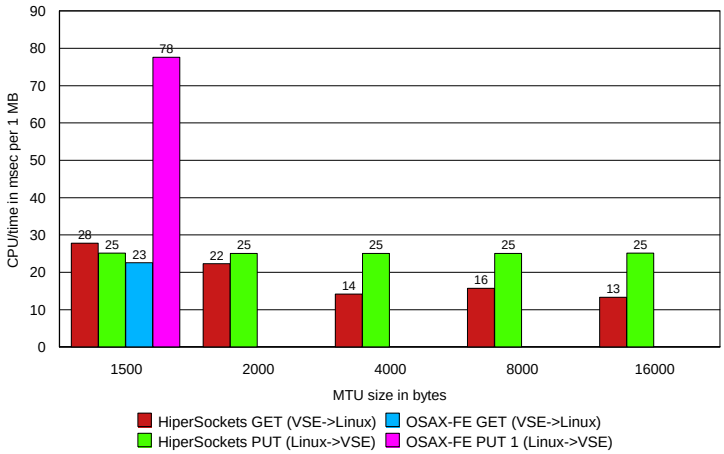
HiperSockets throughput is between 30-80 MB/sec

Throughput (MB/sec) - results (2)



Maximum HiperSockets throughput of 109 MB/sec at 4 concurrent connections

CPU time per MB - results

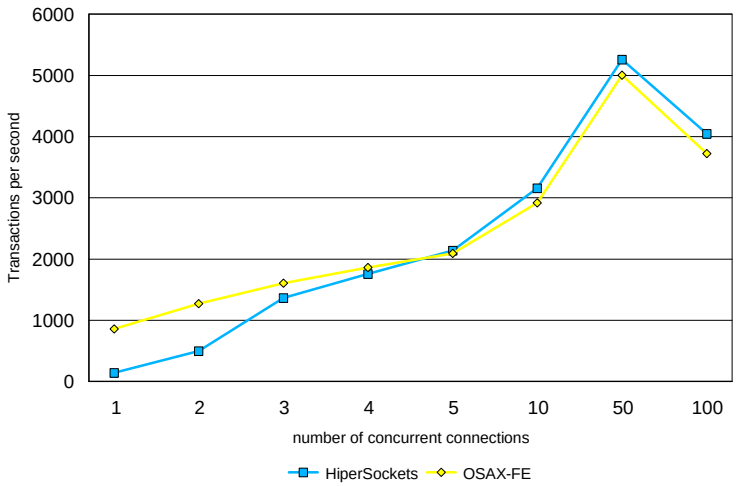


About 15-30 msec CPU time per MB for HiperSockets (on a z800 2066-001)

Transaction per second

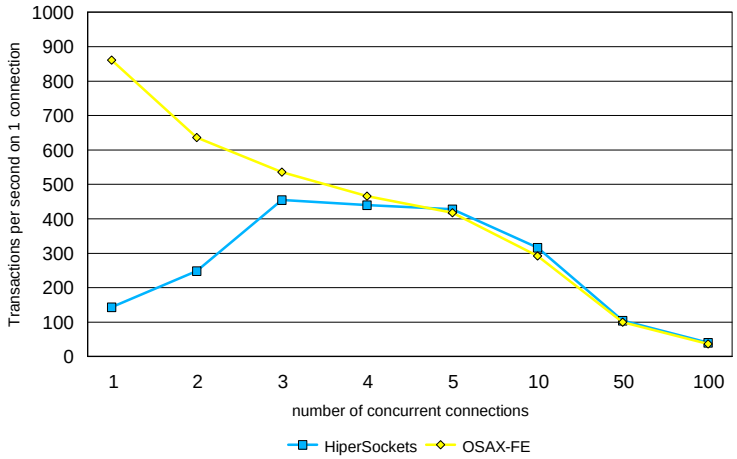
- Measurements has been done with an ECHO server
  - Client on Linux sends 100 bytes to server
  - Server on VSE echoes 100 bytes
  - Per TCP connection 10000 transactions are driven
  - Variations: Number of TCP connections
    - 1,2,3,4,5
    - 10,50,100
  - Measurements
    - Transactions per second
    - CPU time per transaction

Transactions per second - results



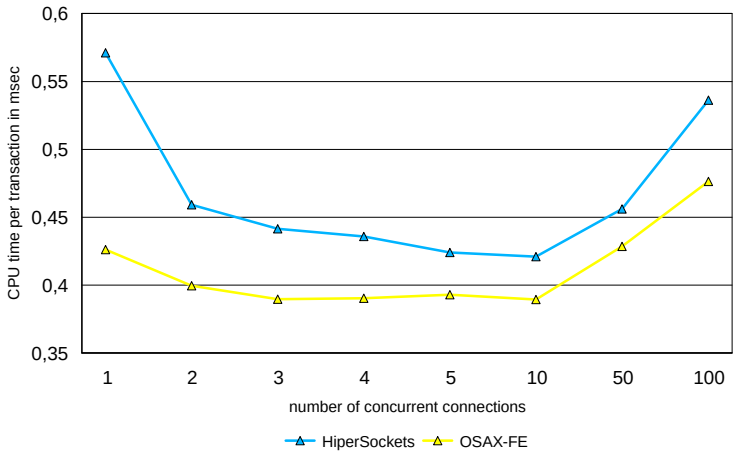
Maximum of 5200 transactions per second at 50 concurrent connections

Transactions per second on 1 connection - results



HiperSockets: Maximum of about 450 transactions per second on 1 connection (= about 2 msec response time)

CPU time per transaction



HiperSockets: About 0.45 msec CPU time per transaction for 2-50 connections

Measurement Results - conclusion

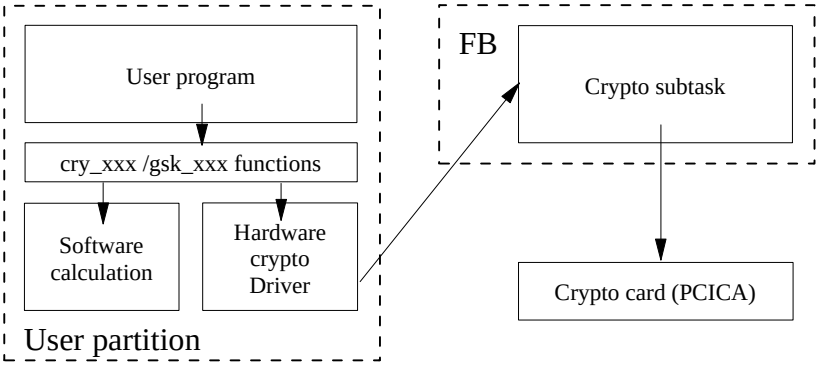
- HiperSockets
  - Throughput
    - Between 30-80 MB/sec
    - Maximum throughput of 109 MB at 4 concurrent connections
    - About 15-30 msec CPU time per MB
  - Transactions per second
    - Maximum of 5200 Transactions per second at 50 concurrent connections
    - About 0.4-0.45 msec CPU time per transaction

Hardware Crypto Overview

- Requires VSE/ESA 2.7 and TCP/IP for VSE/ESA 1.5
- Supported crypto cards
  - PCI Cryptographic Accelerator (PCICA)
    - Feature code 0862
    - Available for zSeries (z800, z900)
- Only RSA (asymmetric) is supported
  - Of benefit for Session initiation (SSL-Handshake)
- Also supported with
  - z/VM 4.2 + APAR VM62905
  - z/VM 4.3

Hardware Crypto Overview - continued

- New crypto subtask in Security Server (SECSERV) running in FB
  - Or as separate job if no SECSERV is running
  - Crypto card is polled by crypto task



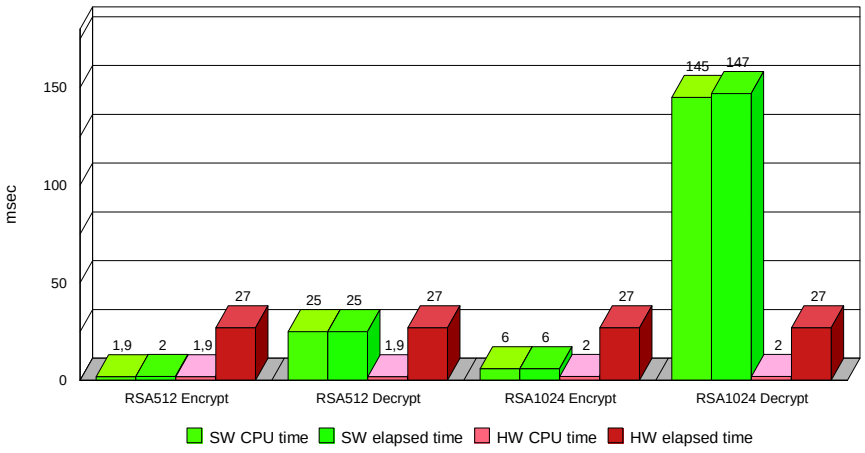
Measurement Environment

- VSE/ESA 2.7 running on a z900 (2064-109)
  - on 1 processor (~2064-101)
  - with a PCI Cryptographic Accelerator
- Testcase programs on VSE
  - Crypto operations measurements
    - calling cry_xxx functions (RSA, DES, SHA, MD5)
    - each crypto operation is performed 10000 times
  - Secured data transfer (SSL)
    - performs SSL handshake
    - performs encrypted data transfer
    - counterpart program running on Windows (SSL-client)
- All RSA operations are measured
  - with Hardware Crypto support
  - with Software Crypto
    - support already available with TCP/IP 1.4/1.5 as shipped in VSE/ESA 2.6

Measurement Environment - continued

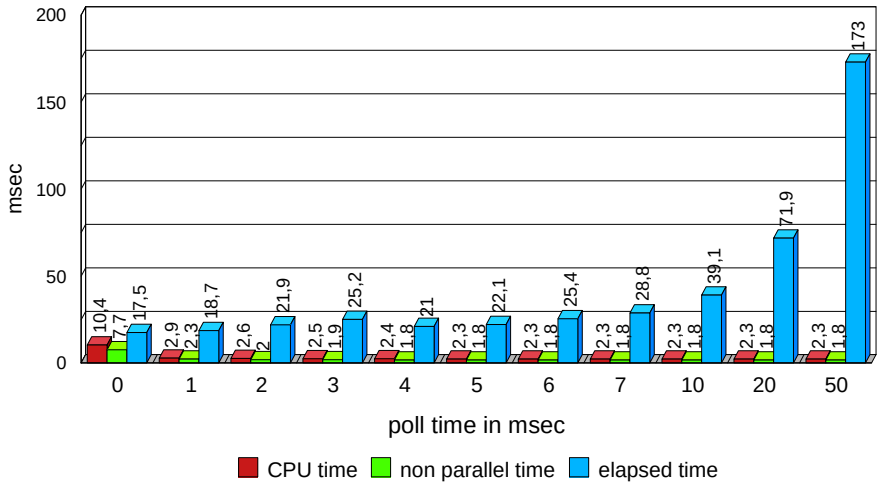
- Variations
  - RSA encrypt/decrypt
    - 512 / 1024 bit key
  - DES, DES CBC, 3DES CBC encrypt/decrypt
    - software crypto only
    - message length (128, 256, 512 bytes)
  - SHA Hash, MD5 Hash, SHA HMAC, MD5 HMAC
    - software crypto only
    - message length (128, 256, 512, 1K, 2K bytes)
  - SSL handshake/data transfer
    - 01 RSA512_NULL_MD5
    - 02 RSA512_NULL_SHA
    - 08 RSA512_DES40CBC_SHA
    - 09 RSA1024_DES_CBC_SHA
    - 0A RSA1024_3DES_EDE_CBC_SHA

Measurements Results - RSA



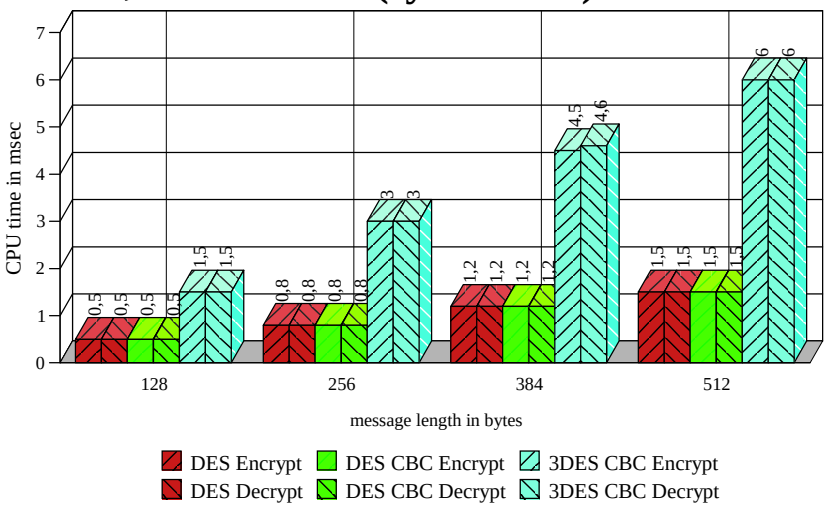
HW Crypto:
- CPU time and elapsed time is independent of operation / key length
- RSA operation takes about 2 msec CPU time and 28 msec elapsed time
- CPU time is always less than software crypto

Measurements Results - RSA polltime



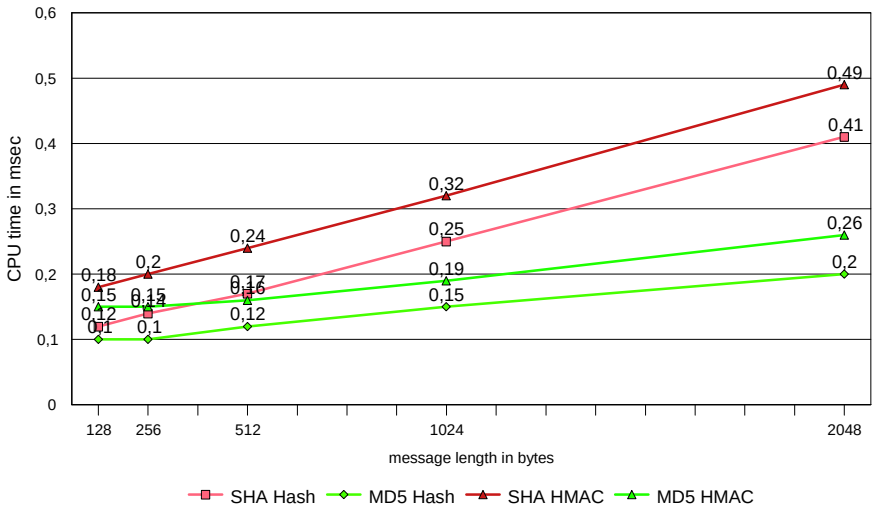
Per default a polltime of 7 msec is used.
Can be changed with: MSG FB,DATA=WAITTIME=nn
Smaller values increase CPU time, higher values increase elapsed time

Measurements Results - DES, DES CBC, 3DES CBC (symmetric)



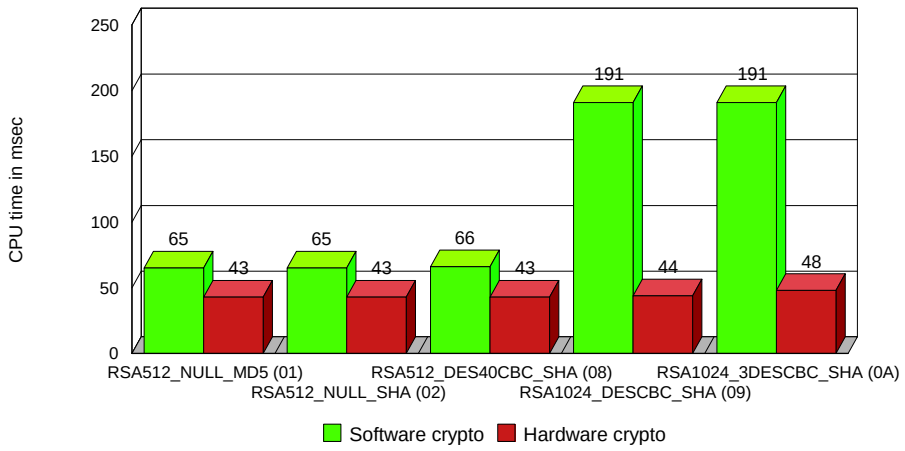
Software Crypto only!
DES and DES CBC takes similar CPU times, 3DES CBC about 3.8 times

Measurements Results - SHA, MD5



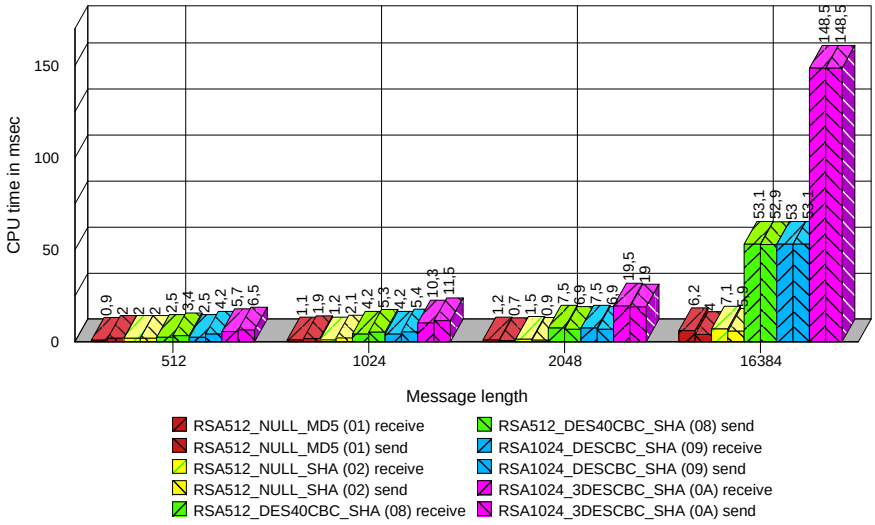
SHA takes about 1.8 times more CPU time compared to MD5
Software Crypto only!

Measurements Results - SSL Handshake



HW Crypto:
- CPU time and elapsed time is independent of cipher suite used
- SSL handshake takes about 43-48 msec CPU time (connection establishment)

Measurements Results - SSL data transfer

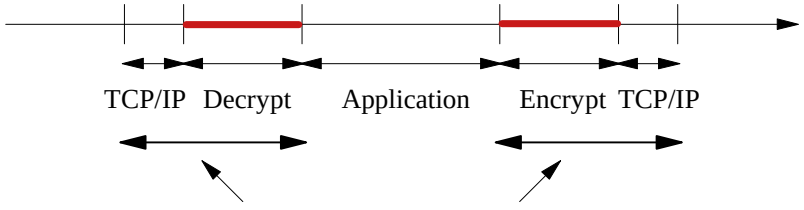


CPU time depends on used hashing (SHA/MD5) and encryption algorithm (DES/3DES)
Software Crypto only!

SSL data transfer overhead

Non SSL

SSL



this has been measured

Measurements Results - conclusion

- HW Crypto
  - Supports RSA operations only (e.g. used by SSL handshake)
  - CPU time/elapsed time is independent of operation and key length
  - Software RSA encryption is faster in terms of elapsed time (on large processors)
    - but hardware crypto saves CPU time
- SW Crypto
  - CPUtime /elapsed time is very dependent on CPU speed and utilization

SSL Performance Recommendations

- Use SSL only if there is a need for
  - If at least one of the following is required
    - Keeping secrets
    - Proving identity
    - Verifying information
- Cipher Suites 01 and 02 has less CPU-time consumption, but NO data encryption
  - RSA512_NULL_MD5, RSA512_NULL_SHA
- If data encryption is required
  - Use cipher suites 08, 09 or 0A
  - 08 uses 512 bit keys, others 1024
  - 1024 bit RSA keylength is recommended (from a security point of view)

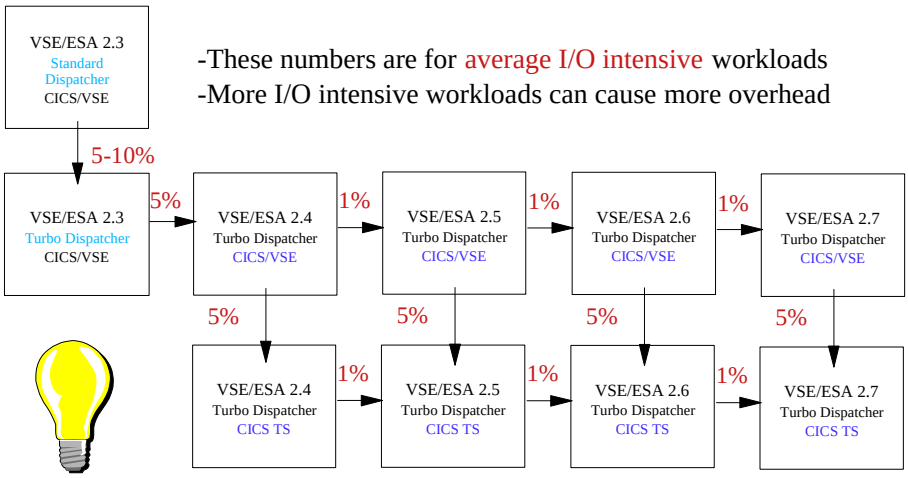
Dependencies for VSE/ESA Growth

- System dependencies
  - Many control-blocks etc.. still below the line
  - VTAM IOBUF areas in System GETVIS-24
  - Non-Parallel-Share limits n-way support
  - Number of tasks
    - Up to 255, 32 per partition, 208 subtasks in total
- Application dependencies
  - Integrated system concepts/functions
  - Functions/Applications dependencies
  - Number of users per TCP/IP partition

Dependencies for VSE/ESA Growth - continued

- Not being considered to be a limit
  - Number of partitions
    - 12 static + 150-200 dyn. partitions
  - Real storage (max. 2 GB)
  - Total virtual storage (max. 90 GB)
  - Total number of devices (3 digit CUU)
    - Max. 1024 devices (and 16 channels)
  - Total number of logical units
    - 255 per partition and 12x255=3060 in total
  - Label area
    - Max. about 9000 in total, and 712 in sub areas

Overhead Deltas for VSE Releases



-These numbers are for average I/O intensive workloads
-More I/O intensive workloads can cause more overhead

Remember that you get a lot of new functions that in most cases helps you to increase VSE system performance and throughput:

Partition Balancing, PRTY SHARE (Turbo Dispatcher), FlashCopy (ESS), Buffer Hasing, Shared data Tables (CICS TS)

Further Information

- VSE Homepage:
  http://www.ibm.com/servers/eserver/zseries/os/vse/
- VSE Performance Homepage:
  http://www.ibm.com/servers/eserver/zseries/os/vse/library/vseperf.htm
- Performance Documents from W. Kraemer
  - ►available on the Performance Homepage
- VSE/ESA e-business Connectors User's Guide
  http://www.ibm.com/servers/eserver/zseries/os/vse/pdf/ieswue20.pdf