IBM Software Group

IBM CICS® Transaction Server for z/OS® V3.1
Performance Report
April 2005

CICS Performance Group
IBM Hursley

ON DEMAND BUSINESS™

© 2007 IBM Corporation

Performance summary (V3.1 compared to V2.3) - Highlights

- **Traditional VSAM (DSW workload)**
  - CPU used - Equivalent
- **Traditional DB2® (RTW workload)**
  - CPU used - Equivalent
- **Java™**
  - CPU used - Equivalent
- **HTTP**
  - Concurrent session limit raised from 900 to 65000 per region
  - Some customers will start using persistent sessions and then also make CPU savings
- **HTTPS**
  - Concurrent session limit raised from 250 to 65000 per region
- **SOAP**
  - Using WSDL2CICS, our workload showed up to 20% CPU saving compared to CTS 2.3 SOAP feature
- **Containers**
  - Program call using CONTAINER, within 2% CPU/Tran of COMMAREA linkage on 1 MIP application
- **XPLINK**
  - Shorter pathlength for C++ method linkage
  - Any benefits will depend on size, number of calls, depth of nested calls etc.
- **OPENAPI**
  - Applications other than those that call DB2 can now exploit OPEN TCBs
  - Throughput not limited by QR TCB, multiple CPUs can now be exploited concurrently.

This foil summarises the following 73 foils and eliminates all the detail!

For Customers migrating traditional CICS/VSAM, CICS/DB2 or Java workloads from V2.3 to V 3.1, they should see no change in CPU utilisation.

Customers using the WEB interface, can now exploit persistent sessions for large networks, as opposed to having to make and break connections with every request. These Customers will make CPU savings. Customers who had small networks and were already able to exploit persistent connections will see up to about 4% increase per on average size applications. They will however benefit from a much greater number of concurrent session per region.

Hursley Performance Environment

- **Exclusive use LPAR on Z990 2084 332**
  - 3 dedicated CPUs for system under test
- **Shared use of LPAR for network simulation**
- **Use LSPR to scale CPU figures to other target systems**
  - 2084 332 ITR/32 or 2084 303 ITR/3 for CP speed comparison?
    - We found the 303/3 to be more accurate for scaling workloads

This describes the Hursley Performance environment. Use the LSPR to scale any CPU figures in this presentation to other processors.

Most of our workloads are currently driven by TPNS which simulates Network Devices, either LU2 or TCPIP.

The system under test resides on one LPAR and TPNS on an other to avoid interference with the measurement. RMF is used to measure performance at a system level and address space level. If further analysis is required we have tools on these systems for attributing CPU usage down to Modules and CSECTS within Modules.

Release/Release

Release/Release

ON DEMAND BUSINESS™

Release/Release Comparisons

- **COBOL VSAM**
  - 'DSW ' workload
- **COBOL DB2**
  - 'RTW' workload
- **Java**
  - Standard Java/JCICS and EJB workload
- **Compare ITR (tran per CPU second)**
  - Transaction rate in TORs/Total CICS regions CPU
- **Taken from average of 5 Transaction rates**

Release/Release measurements are those aimed at understanding if any of the code changes we have made have altered the performance characteristics of original function. i.e. will it affect customers who migrate workloads, without using any of the new functions.

IBM Software Group

Release to Release - DSW Workload (VSAM)

- **Hardware**
  - 2084-303 system under test
  - 2084-303 TPNS driver
- **CICS Environment**
  - 2 TORs, 2 AORs, 1 FOR all MRO connected
  - All VSAM files recoverable
  - VTAM HPO used
  - No transaction isolation or storage protection
  - Long running mirror in FOR
- **Software**
  - z/OS 1.6
- **Workload**
  - 34 transaction types
  - COBOL applications
  - 40% transactions invoke menu
  - 32 VSAM KSDS files
  - average 6 FC calls per transaction
  - EXEC CICS LINK to program for each FC call
  - 69% Read, 10% Read for Update, 9% Update,
  - 11% Add , 1% Delete
  - XEIIN and XEIOUT enabled
  - TRUE enabled
  - CICS Performance Monitoring on

The DSW is our standard workload for looking at VSAM type workloads.

Nowadays it has a comparatively low pathlength compared to many Customer applications. In some ways this is good because any changes in CICS code is not masked by large chunks of application code.

IBM Software Group

DSW Performance Data

Average ITR = 2809

ETR	TORs%	AORs%	FOR%	Resp	ms/tran	ITR
384.99	7.9	23.6	10.9	0.037	1.10	2727
531.50	10.6	32.2	14.8	0.036	1.08	2777
708.11	14.2	42.4	19.0	0.039	1.06	2830
1055.21	20.9	62.7	28.5	0.036	1.06	2830
1586.01	31.6	92.8	41.5	0.040	1.04	2884

V2.3

Average ITR = 2794
Delta = 0.5%

ETR	TORs%	AORs%	FOR%	Resp	ms/tran	ITR
384.93	8.0	23.9	11.0	0.037	1.11	2702
531.49	10.8	32.5	14.9	0.036	1.09	2752
710.50	14.2	43.1	19.3	0.035	1.07	2803
1053.94	20.8	63.2	28.5	0.037	1.06	2830
1564.69	30.8	92.0	40.7	0.043	1.04	2884

V3.1

7 IBM CICS TS for z/OS V3.1 | Performance Report ON DEMAND BUSINESS™

ETR is the total number of transactions per second accumulated across the 2 TORs.

TOR% is the total CPU used for the 2 Terminal Owning Regions expressed as a percentage of the elapsed time. This is reported in RMF at the RPGN level by APPL%. Based on CPU processor seconds therefore can exceed 100%.

AOR% same definition as TOR% but for the 2 Application Owning Regions.

FOR% same definition as TOR% but for the File Owning Region.

Resp is the CICS internal response time.

ms/tran is the average number of milli seconds of CPU per transaction

ITR is the ETR/CPU%. A drop in ITR means an increase in pathlength and loss of throughput capability. Transactions per CPU second

Release to Release - RTW Workload (DB2)

- **Hardware**
  - 2084- 303 system under test
  - 2084- 303 TPNS driver
- **CICS Environment**
  - Single Region
- **DB2 Environment**
  - 7.1.0
  - Thread priority = High
  - Protect num = 0
  - Account Rec = UOW
  - SMF 89 not collected (usage pricing)
  - No Class (1) accounting
- **Software**
  - z/OS 1.6
- **Workload**
  - 7 transaction types
  - COBOL applications - Not Thread safe
  - 20 Database tables
  - Average 200 DB2 calls per transaction
  - 54% Select, 1% inset, 1% update, 1%delete,
  - 8% open cursor, 27% fetch cursor 8 close cursor
  - No CICS tracing

The following foils show the performance data for the RTW workload. A COBOL/DB2 application which in it's standard form was run as NON-THREADSAFE.

Of course if you are able to migrate your applications from CTS 1.3 and specify them as Threadsafe then you will see performance benefits because the switching back and forward between TCBs will be eliminated. The more DB2 calls per application, the more the saving.

IBM Software Group

RTW Performance Data

Average ITR = 427

ETR	CPU%	Resp	ms/tran	ITR
41.49	29.3	0.032	7.06	425
62.22	42.5	0.024	6.83	439
99.59	67.5	0.016	6.78	443
245.08	176.5	0.037	7.20	417
315.65	230.7	0.081	7.31	410

V2.3
DB2 7.1

Average ITR = 426
Delta = 0.2%

ETR	CPU%	Resp	ms/tran	ITR
41.5	28.6	0.017	6.89	435
62.64	42.8	0.017	6.83	439
99.56	69.0	0.018	6.93	433
245.78	177.9	0.039	7.24	414
286.88	210.4	0.059	7.33	409

V3.1
DB2 7.1

9 IBM CICS TS for z/OS V3.1 | Performance Report ON DEMAND BUSINESS™

ETR is the total number of transactions per second

CPU% is the total CPU used. Based on CPU processor seconds therefore can exceed 100% of a 3 way.

Resp is the CICS internal response time.

ms/tran milliseconds of CPU per transaction

ITR is the ETR/CPU%. A drop in ITR means an increase in pathlength and loss of throughput capability. Transactions per CPU second

Release to Release - Java Workloads

- **Hardware**
  - 2084-303 system under test
  - 2084-303 TPNS driver
- **CICS Environment**
  - Single Region
  - Java JDK™ 1.4.2
  - Single JVM™
  - REUSE=RESET and REUSE=YES
- **Software**
  - z/OS 1.6
- **JCICS Workload**
  - Simple Java application
  - Ave. 100 mixed API calls
  - 3270 driven
- **EJB workload**
  - Simple EJB™
  - Uses JCICS to call COBOL 'backend'
  - COBOL application is about 3.4ms of CPU

To give us an indicator as to whether or not the performance of Java has changed from one release to the next we used two different workloads run in a variety of Java configurations.

The JCICS workload makes an combination of JCICS API calls and is driven via an LU2 TPNS script.

The EJB workload is also driven using TPNS but this time over TCPIP. The EJB in CICS makes a call using JCICS to a simple COBOL application program and then returns a String to the client.

These workloads were run using the RESET, CONTINUOUS and SHARED JVMS.

IBM Software Group

JCICS Performance Data

ETR	CPU%	Resp	ms/tran	ITR
66.20	24.8	.009	3.74	802.1
98.57	36.6	.013	3.71	808.6
139.75	51.6	.015	3.69	813.0
193.76	71.7	.016	3.70	810.8
254.98	94.2	.092	3.69	813.0

V2.3 REUSE=RESET

ETR	CPU%	Resp	ms/tran	ITR
66.26	21.3	0.007	3.21	934.5
99.19	32.1	0.008	3.23	928.7
141.41	44.6	0.007	3.15	952.3
195.43	61.4	0.010	3.14	955.4
297.27	93.6	0.037	3.14	955.4

V2.3 REUSE=YES

ETR	CPU%	Resp	ms/tran	ITR
66.23	25.1	0.008	3.78	793.6
98.70	37.1	0.011	3.75	800.0
139.69	52.3	0.014	3.74	802.1
193.49	72.5	0.016	3.74	802.1
251.92	94.3	0.097	3.74	802.1

V3.1 REUSE=RESET

ETR	CPU%	Resp	ms/tran	ITR
66.43	21.6	0.008	3.25	923.0
99.32	32.3	0.008	3.25	923.0
141.40	44.6	0.007	3.15	952.3
197.07	61.8	0.007	3.13	958.4
298.36	94.1	0.035	3.15	952.3

V3.1 REUSE=YES

ON DEMAND BUSINESS™

Average ITR for V2.3 REUSE=RESET is 809.5

Average ITR for V3.1 REUSE=RESET is 799.9 approx 1% delta

Average ITR for V2.3 REUSE=YES is 945.2

Average ITR for V3.1 REUSE=YES is 941.8

Notice the performance benefit from using REUSE is from not reseting the JVM at the end of a transaction.

EJB Performance Data

ETR	CPU%	Resp	ms/tran	ITR
83.34	43.9	0.007	5.26	570.3
95.12	50.1	0.007	5.26	570.3
110.38	58.1	0.008	5.26	570.3
131.73	69.2	0.010	5.25	571.4
185.50	97.3	0.028	5.24	572.5

V2.3 REUSE=RESET

ETR	CPU%	Resp	ms/tran	ITR
83.43	33.9	0.005	4.06	738.9
95.13	38.6	0.005	4.05	740.0
110.6	44.7	0.005	4.05	740.0
132.16	53.4	0.006	4.04	742.5
188.86	76.1	0.008	4.02	746.2

V2.3 REUSE=YES

ETR	CPU%	Resp	ms/tran	ITR
83.32	43.4	0.007	5.19	578.0
95.12	50.4	0.007	5.29	567.1
110.38	58.5	0.009	5.29	567.1
131.73	69.7	0.011	5.29	567.1
185.21	97.8	0.029	5.28	568.1

V3.1REUSE=RESET

ETR	CPU%	Resp	ms/tran	ITR
83.16	34.5	0.007	4.14	724.6
95.19	39.4	0.006	4.13	726.3
110.49	45.7	0.006	4.13	726.3
132.15	54.6	0.006	4.13	726.3
188.79	77.7	0.009	4.11	729.9

V3.1 REUSE=YES

Average ITR for V2.3 REUSE=RESET is 570.9

Average ITR for V3.1 REUSE=RESET is 569.4

Average ITR for V2.3 REUSE=YES is 741.8

Average ITR for V3.1 REUSE=YES is 726.6, a delta of about 2%

IBM Software Group

HTTP

HTTP

ON DEMAND BUSINESS™

HTTP - Performance perspective

- **Increase concurrent connections to 65000**
  - No affinity to long running CWXN transactions
- **3.1 Customers with large networks**
  - Can now use keep-alive and SocketClose (NO)
  - Now have a new CWXN for each request
  - No need to make and break the connections
- **Migrating from a keep-alive (no) environment**
  - Increased session limit and less CPU per request
- **Migrating from a keep-alive (yes) environment**
  - Increased session limit, +3 to +4% CPU on a 2.5 MS transaction

ON DEMAND BUSINESS™

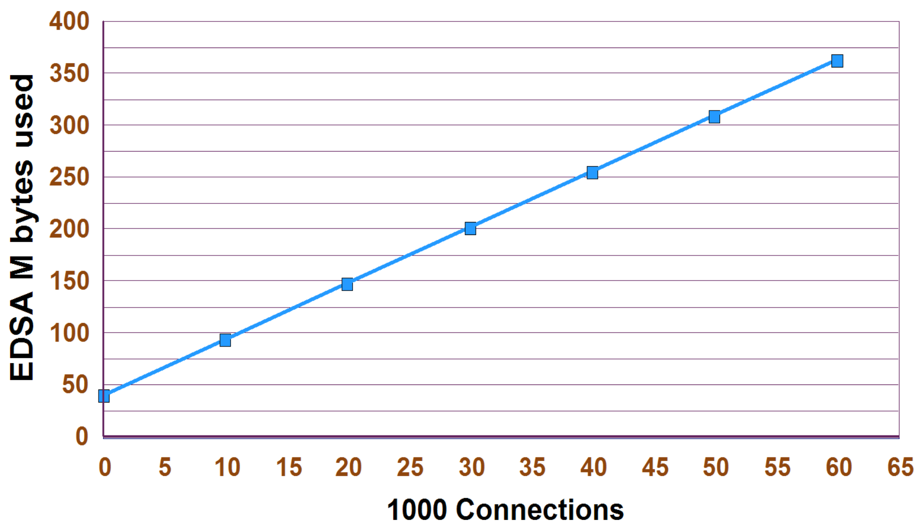
Prior to V3.1, every persistent HTTP connection had affinity to a CWXN transaction which existed for the life of the connection. This limited the number of potential concurrent connections to MXT.

So if there was a requirement to support a large network, keepalive=no or socketclose(0) needed to be specified. This meant that for each request a new connection was opened and then closed.

V3.1 supports up to 65000 concurrent connections in any one region. This means that large networks can be migrated and connections changed from non-persistent to persistent, saving the cost of opening and closing the connection for each request.

IBM Software Group

Inbound HTTP - Storage per Connection



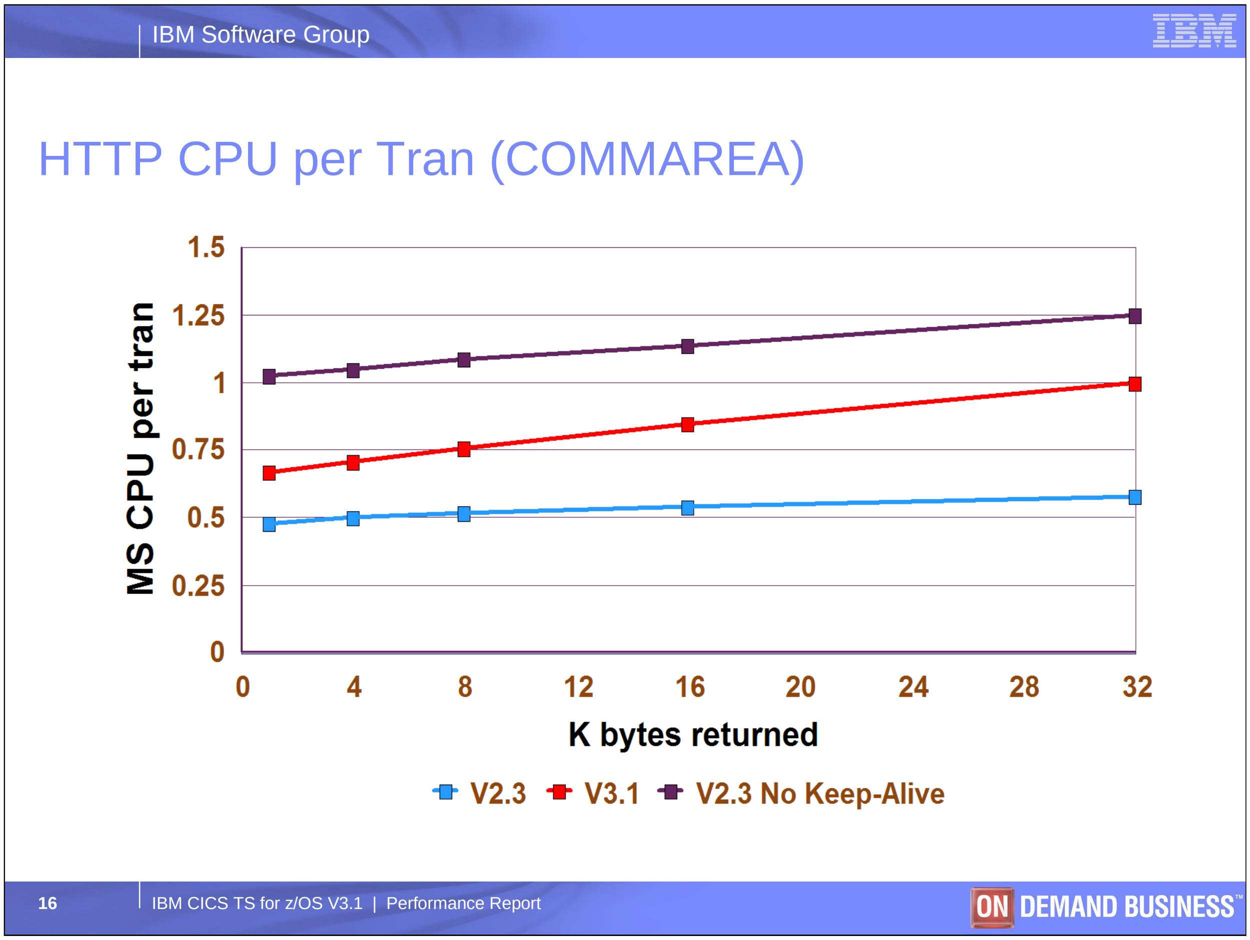
Approx. 4.8k per connection

This chart shows the CICS V3.1 EDSA storage used as the number of inbound connections in the CICS region increases.

Excluding any application data that may be flowing in and out, this shows that cost of a single connection to be about 4.8 K above the line storage.

The max number of inbound connections supported is 65000

The max number of outbound connections supported is 32000

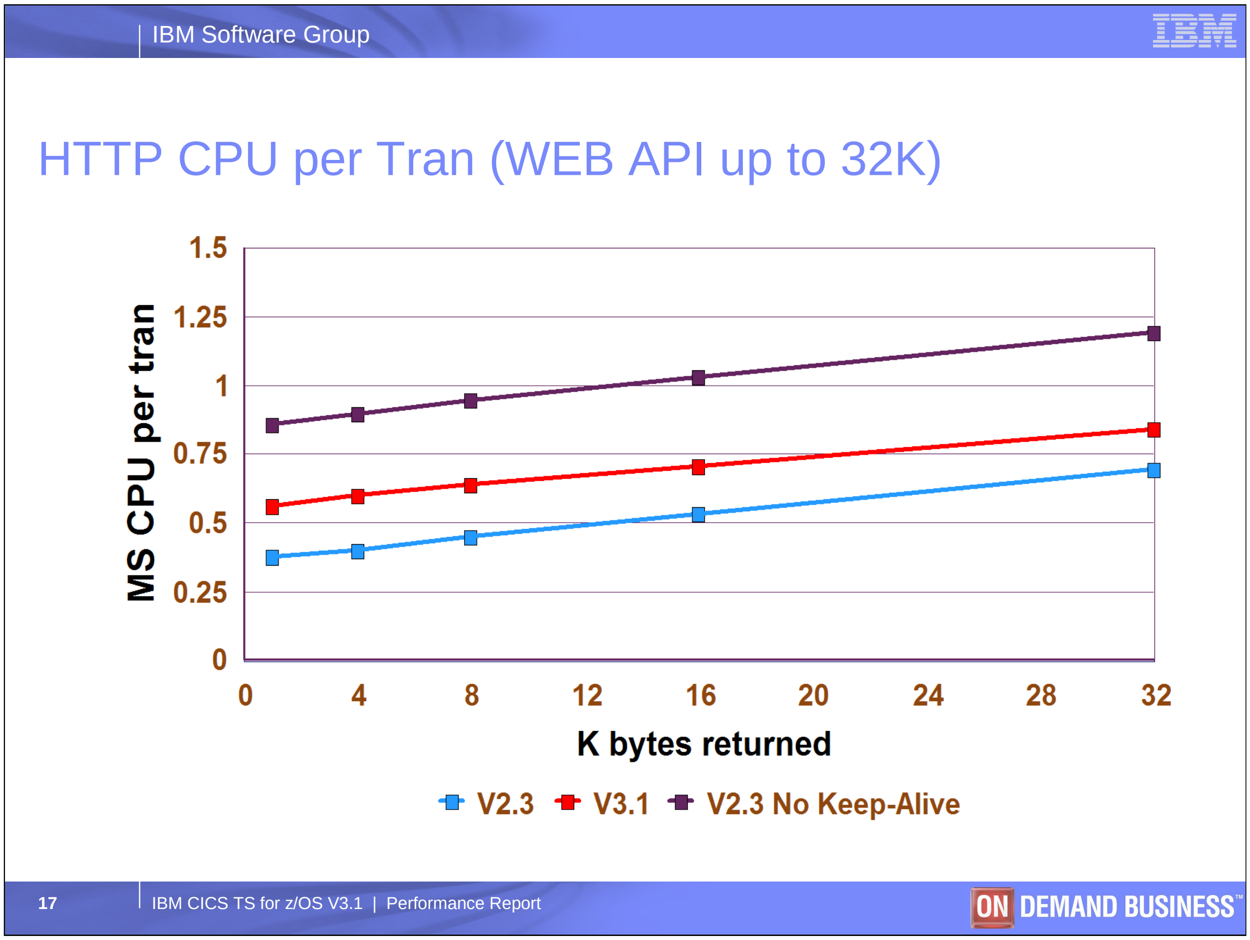


The following charts aim to show CPU costs required to support the CICS/HTTP infrastructure. The target application, is relatively small and simply returns various sizes of data to the Browser.

This chart shows the CPU per 'business' transaction. I.e.. the CPU used, including the TCPIP listener CSOL, CWXN and CWBA. The application simply receives a fixed length data which contains the length of data structure to be returned to the Browser.

COMMAREA applications have a maximum of 32K that can be sent and received.

The chart also shows the cost of non-persistent connections in V2.3.



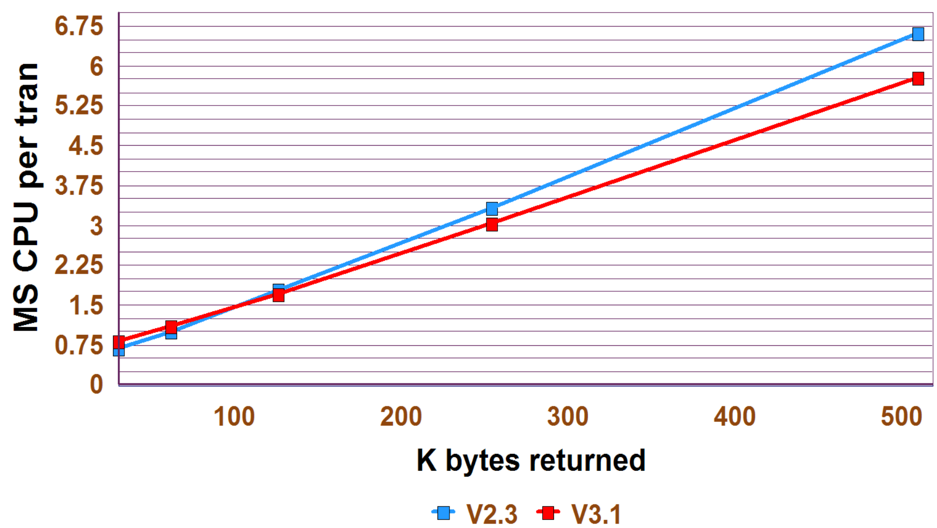
This chart shows the CPU per 'business' transaction. I.e.. the CPU used including the TCPIP listener CSOL, CWXN and CWBA. The application simply receives a fixed length data which contains the length of data structure to be returned to the Browser.

This application used the WEB API instead of the COMMAREA interface shown in the previous chart.

It is shown here for completeness and although it does demonstrate a lower CPU per transaction than the COMMAREA application program, not too much should be read into this as these are very small assembler programs and sensitive to data alignment etc.

IBM Software Group

HTTP CPU per Tran (WEB API over 32k)



Includes TCPIP

ON DEMAND BUSINESS™

Applications using the WEB API can xmit data greater than 32K.

This chart shows that as the data size increases, the CPU per transaction used in V3.1 starts to decrease in comparison to V2.3. This was due to a code change in the EXEC CICS INSERT DOC which gave a performance improvement in V3.1

HTTPS

HTTPS

HTTPS - Terms

- **Full handshake**
  - Negotiate session parameters, cipher suite etc.
  - Done once per connection by CWXN for a new connection
- **Partial handshake**
  - Done when client has previously had an SSL connection that has since been closed
    - And the Client has retained SSL session id
    - The Session id has been retained in CICS storage or Coupling Facility
  - Done within the CWXN transaction
- **Encrption/Decryption**
  - For a persistent connection only the one handshake is needed
  - After that just Encyption/Decryption
  - Done by the CWBA transaction

ON DEMAND BUSINESS™

HTTPS - Terms

- **Hashing**
  - Ensure data integrity during transport (ie. not been changed),
  - Two common algorithms
    - MD5 Done in software
    - SHA-1 Done in Hardware
- **CPACF**
  - CP Assist for Cryptographic Function
  - Available on every CP so no affinity issues
  - SHA-1 TDES and DES Encypt/decrypt done by these on Z990
- **PCIXCC**
  - Peripheral Component Extended Cryptographic Coprocessor
  - Full Handshakes are done by these on Z990 if ICSF active
  - Handsheke rates and card Utilisation are recorded by RMF

System SSL running on a 2084 system with the CP Assist facility (CPACF) will exploit the capabilities provided in the following ways

z/OS V1.4 and z/OS V1.5 will exploit the DES and TDES capabilities in the CPACF, if ICSF is up and running prior to the start of the System SSL application (i.e.. CICS)

z/OS V1.6 will exploit DES, TDES and SHA-1 in the CPACF. It will exploit this capability whether or not ICSF has been started.

In all cases, the calls to the CPACF are done directly to the CPACF by System SSL. ICSF is used in 1.4 and 1.5 to determine whether the CPACF is installed. In 1.6, System SSL determines this itself.

RC2 and RC4, encrypts and decrypts are always done in software.

MD5 Hashing is always done in Software

SHA Hashing is done in the Hardware

HTTPS - Performance perspective

- **Increase concurrent connections**
  - No affinity to SSL TCB for duration of connection
    - SSL TCB per connection now not required
  - No affinity to long running CWXN transaction
- **Easy choice over Cipher suites**
  - Specified on TCPIPSERVICE definition
- **Sysplex wide cache for SSL session id**

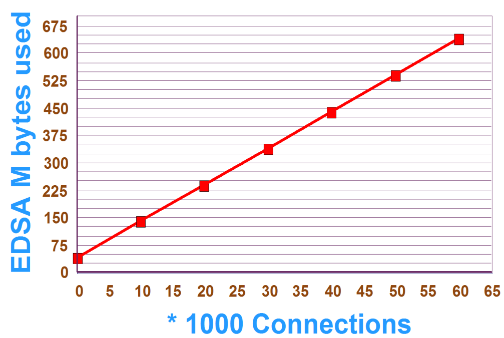
ON DEMAND BUSINESS™

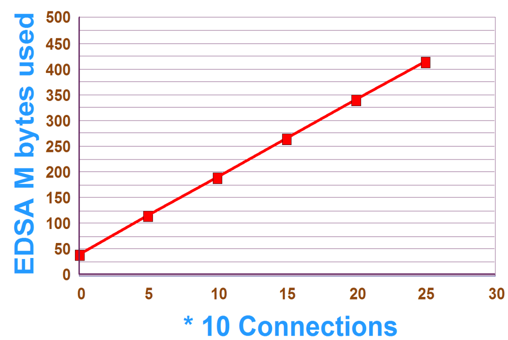
Prior to V3.1 an SSL connection, not only had affinity to the CWXN as described previously in the HTTP case, but each connection required exclusive use of an SSL TCB and all the storage associated with it for the duration of the session. V3.1 removes this requirement and therefore allows many more concurrent connections.

V3.1 allows the Cipher suite to be used by the session to be specified on the TCPIP Service definition. The choice of Cipher Suite can be performance sensitive as shown later.

Also V3.1 supports central caching of SSL session ids in the Coupling facility. This means the session id can be reused even if the session is routed to another CICS region in the Sysplex after a reconnection.

HTTPS - Storage per Connection





V3.1
Max. = 65000 connections
Approx. 10K EDSA + 3K MVS storage per connection

V2.3 - physical restriction was 17K below line per connection
+ 1.5 M above per connection
+ I TCB per connection
Max. = 250 connections

ON DEMAND BUSINESS™

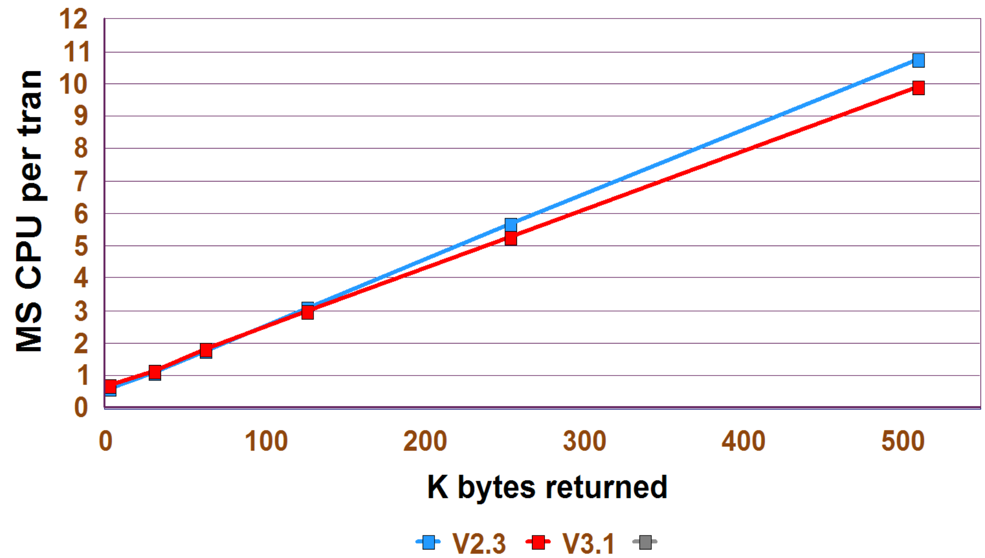
Charts show the number of connections and the CICS storage usage, excluding any application data that is being flowed.

Charts show the decreased use of storage in V3.1 compared to V2.3.

Also with V2.3, each connection has an affinity to a dedicated TCB for its lifetime so if you needed 200 connections, that required 200 dedicated TCBs.

In V3.1 SSL TCBs are only used during encryption, decryption and hanshaking and are reused by different connections.

V2.3 Vs V3.1 HTTPS
- DES 56 Encryption / Decryption costs



Data taken from RMF, includes CWXN and CWBA

On V2.3 it is difficult to control which Cipher suite to use for an SSL connection.

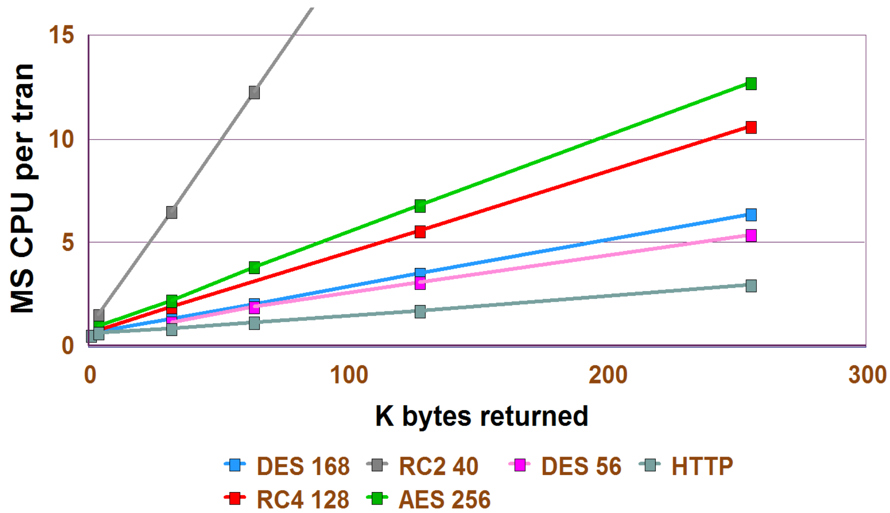
By default V2.3 was using DES 56 for this workload so to make a Release/Release comparison we configured the V3.1 TCPIPSERVICE so that it also used DES 56 for all connections.

This chart shows the CPU per transaction comparison for various sizes of data being flowed.

This workload was using persistent connections in both releases so data only shows encryption and decryption costs, no handshaking costs.

Of course the increased connection support in V3.1 may enable some customers to migrate their network and change from non persistent to persistent connection and therefore eliminate the cost of opening/closing connections and SSL handshaking costs.

HTTPS – Encryption / Decryption costs



SSL_RSA_WITH_3DES_EDE_CBC_SHA (approx. 0.022 ms CPU per 1k)
SSL_RSA_WITH_RC4_128_SHA (approx. 0.038 ms CPU per 1k)
SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5 (approx. 0.180 ms CPU per 1k)
SSL_RSA_WITH_DES_CBC_SHA (approx. 0.015 ms CPU per 1k)
TLS_RSA_WITH_AES_256_CBC_SHA (approx. 0.046 ms CPU per 1k)

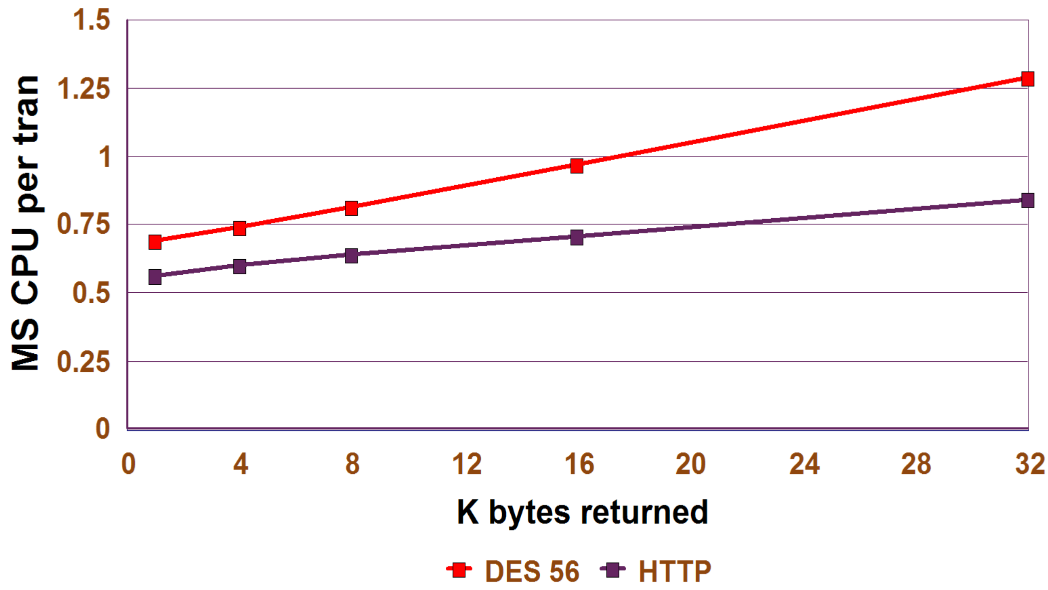
This chart aims to show a comparison cost in terms of CPU for some of the various Cipher Suites supported by CICS V3.1

Persistent connections were used and measurements were taken at a steady state. No SSL Handshaking costs involved.

The data in this chart can be used for capacity planning exercises, if you know which cipher suite you are going to use, you can roughly calculate what the costs will be by multiplying the length in K by the above costs.

IBM Software Group

HTTPS (DES 56) vs HTTP Encryption/Decryption costs



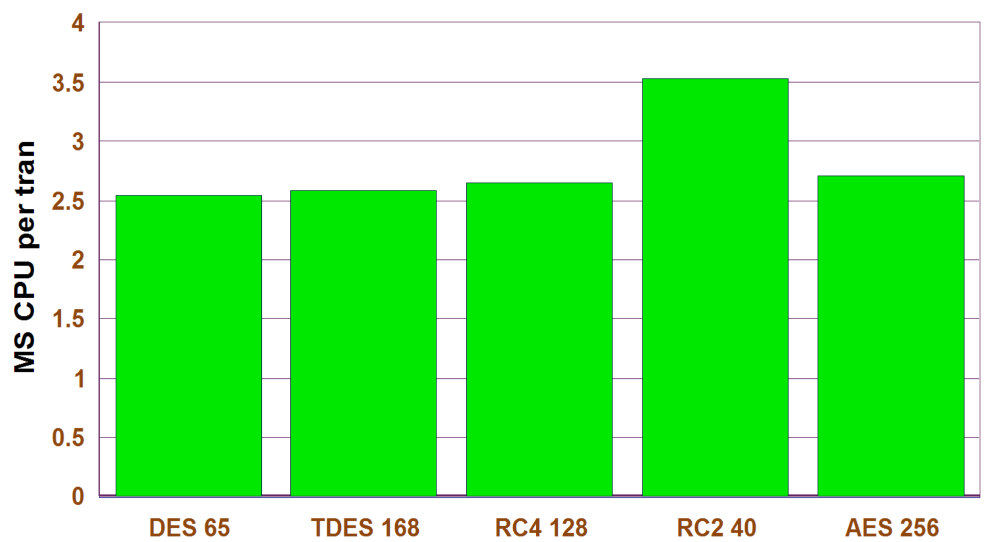
26 IBM CICS TS for z/OS V3.1 | Performance Report

ON DEMAND BUSINESS

This chart shows the CPU per 'business' transaction. I.e.. the CPU used including the TCPIP listener CSOL, CWXN and CWBA. The application simply receives a fixed length data which contains the length of data structure to be returned to the Browser.

The aim here is to show a quick comparison between HTTP and HTTPS when data is flowed over persistent connections.

SSL Full Handshake CPU costs



Includes CSOL, CWXN and CWBA CPU costs

ON DEMAND BUSINESS™

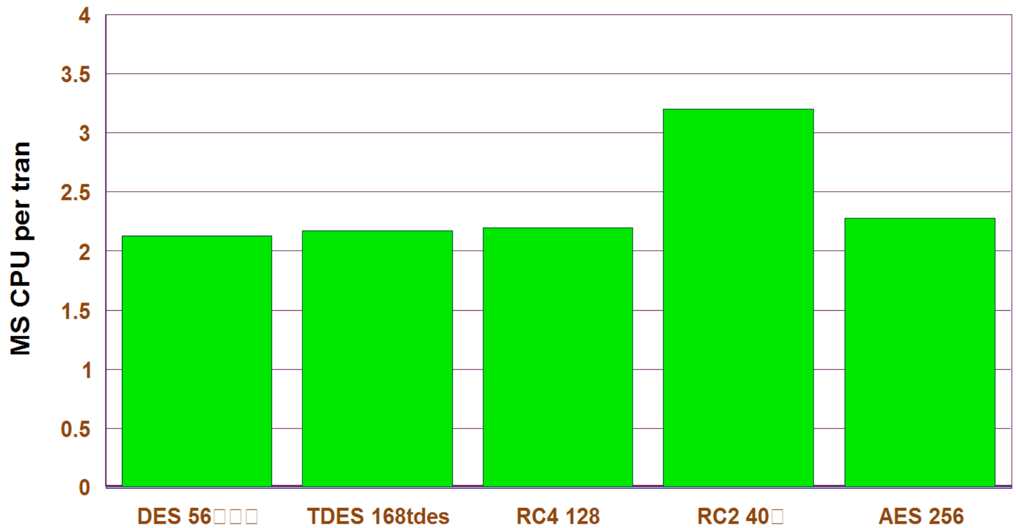
A full handshake is done when a client connects for the first time or on subsequent connects if the SSL session id has expired and is therefore not available for reuse. This expiry is controlled by SSLDELAY. After a session closes, the session id remains available for reuse, in a CICS or Coupling Facility Cache for the SSLDELAY duration.

Data includes Encryption and decryption costs for 4K data flows.

Full handshakes are executed on the Peripheral Component Extended Cryptographic Coprocessor and this is reported via RMF

-------- CRYPTOGRAPHIC COPROCESSOR --------					
		-------- TOTAL --------			KEY-GEN
TYPE	ID	RATE	EXEC TIME	UTIL%	RATE
PCIXCC	0	272.4	0.8	21.9	0.00
	1	0.0	0.0	0.0	0.00

SSL Partial Handshake CPU costs



Includes CSOL, CWXN and CWBA CPU costs

ON DEMAND BUSINESS™

Partial handshakes are done when a client connects for the 2nd and subsequent times and there is a valid SSL session id available to reuse.

The SSL session id can either be stored in the CICS cache which means it will only reuse if it goes back to the same CICS region. SSLCACHE=CICS

Or the SSL session id can be stored in the Coupling Facility, in which case it can return to any CICS region in the Sysplex and reuse the previously negotiated SSL session id. SSLCACHE=SYSPLEX

Data includes Encryption and decryption costs for 4K data flows.

Open API

OPEN API

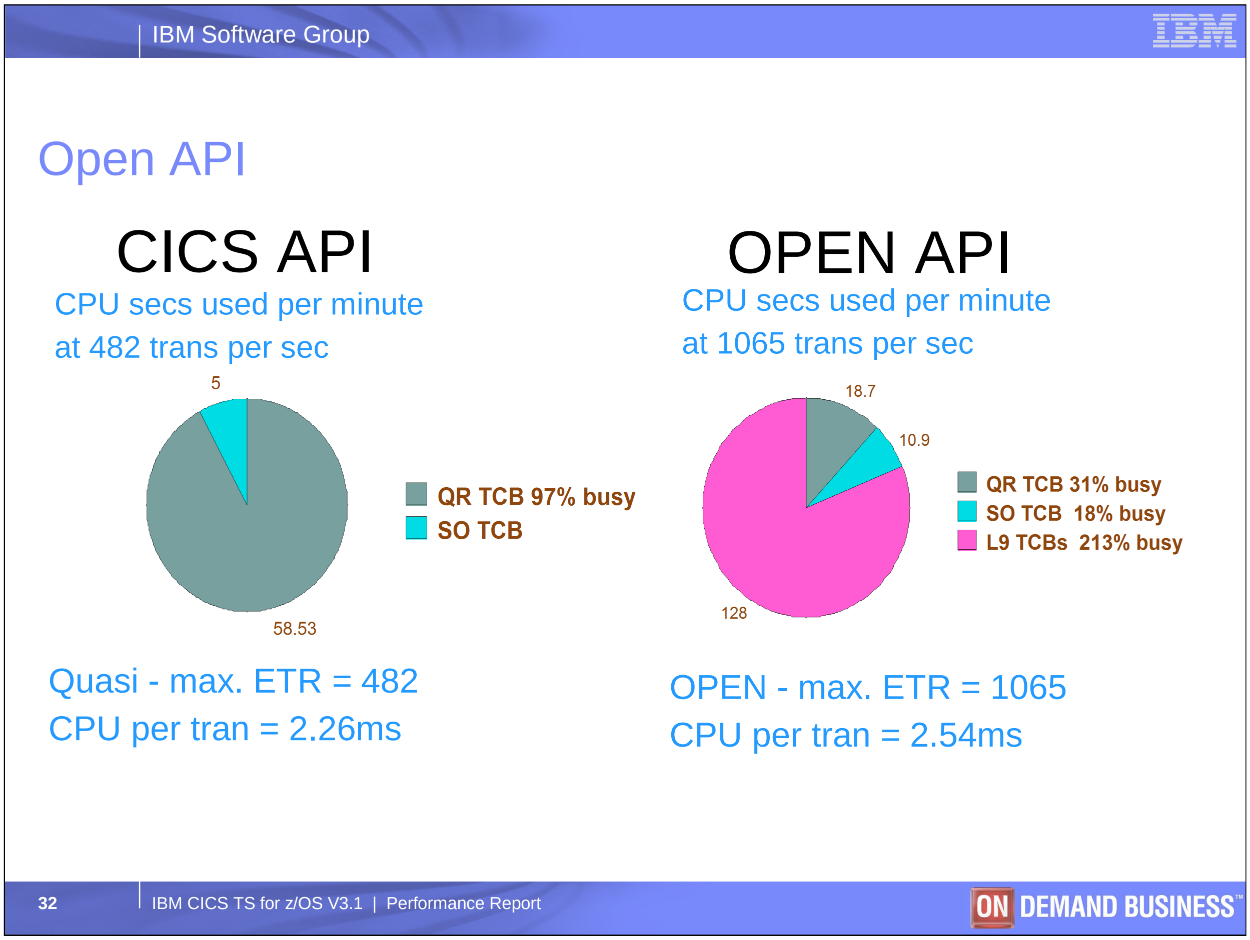
Open API

- **Running on 'Open TCBs' gives CPU concurrency**
- **Throughput is not limited to the speed of a single CP**
- **Reduces any contention for the 'QR' TCB**
- **Open API is not intended to reduce CPU per transaction**
- **It does give more potential throughput per region**
- **Allows the application to use other APIs**
  - Those that would otherwise 'block' the QR TCB

Open API

- **Prior to V3.1**
  - Threadsafe DB2 applications ran on L8 Open TCBs after a DB2 call
- **V3.1**
  - Threadsafe and CICSAPI or OPENAPI on program definition
  - OPENAPI puts program on L8 or L9 during initialisation
  - USER key = L9 CICS key = L8

ON DEMAND BUSINESS™



This chart shows the effect of changing a WEB interface program to OPENAPI.

The workload was driven until no more throughput could be achieved.

With the CICSAPI version, at 482 transaction per second, the QR TCB became the constraining factor at 97% busy.

When using OPENAPI, we achieved 1065 transactions per second and the limiting factor was overall CEC CPU.

Note that the acutal CPU per tran has risen and this can happen due to CPU CACHE miss rate increasing as more CPUs are used concurrently and code has more chance of being dispatched on a CPU other than the one it was previously using.

Open API Cautions!

- **Choose applications carefully**
  - Avoid applications with non-threadsafe APIs
    - e.g.. lots of File Control (this will cause TCB switching back to QR)
- **Migrating Threadsafe DB2 applications from V2.3**
  - userkey/threadsafe on V2.3 ------------------- application runs on L8
  - userkey/threadsafe/OPENAPI on V3.1 ----- application runs on L9
    - DB2 calls then need to switch to L8!
  - Leave DB2 Threadsafe applications moving from V2.3 as CICSAPI!

XPLink

XPLINK

ON DEMAND BUSINESS™

XPLINK

- **XPLINK introduced in OS/390 V2.10**
  - feature of z/OS that provides high performance subroutine call and return mechanism
  - Supported by C/C++ compiler of z/OS
  - Specified by C/C++ compiler option XPLINK
  - Previously not supported in CICS
- **XPLINK requires MVS LE rather than CICSLE**
  - Therefore application requires own TCB to run on
- **XPLINK programs execute like OPENAPI programs**
  - Runs on an X8 or X9 TCB with MVS/LE rather than L8 or L9 with CICS/LE
  - CICS detects at runtime that program compiled with XPLINK

XPLINK cautions!

- **XPLINK programs run on X8 or X9 TCBs**
  - Mixing DB2 calls will cause switches to L8 TCBs
  - Might be better not using XPLINK but C++ with THREADSAFE CICSAPI for DB2
- **Mixing lots of non_threadsafe CICS API will cause excessive switching to QR TCB**
- **Foundation Classes are currently not XPLINK compiled**
  - A call to one of these will cause a TCB switch

ON DEMAND BUSINESS™

IBM Software Group

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

CICS DB2 IBM OS/390 RMF VTAM z/OS

EJB, Java, JDK, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance 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 or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.

ON DEMAND BUSINESS™