

CICS TS V5.5 Performance Highlights



Jenny He

hejen@uk.ibm.com

CICS development, IBM Hursley Lab, UK

November 2019

Session GE



Notes

Every second slide in this presentation is a notes slide like this one and provides a background on the previous slide's content.

Not all main presentation slides require an accompanying notes slide, however one is always provided to maintain the even / odd numbering scheme.

Session abstract

The CICS TS V5 releases include many performance improvements to increase horizontal and vertical scalability. This session highlights some of these enhancements, combining reductions in storage and CPU usage with extra monitoring data available for all types of applications. In this session, we look at recent improvements to MRO, web services, Java, zIIP eligibility, and even new commands, helping you save money and improve throughput.

Measurement process

Notes

This section looks at how we assess the performance of the CICS product.

Measurement process

- Overnight automation on dedicated LPAR
 - Dedicated CPUs, CHPIDs, DASD
- 5 RMF intervals recorded
 - Various transaction rates
- Total CICS address space accumulated
 - Divided by transaction rate to give CPU/tran
- Average CPU/transaction over 5 intervals compared
- Any difference analysed using Hardware Instrumentation (HIS)

Notes

Hardware Instrumentation Service is a facility that has been available in IBM Z hardware for several generations. It is a sampling process which allows detailed inspection of where CPU hot-spots lie when running a workload.

Environment for V5.5 performance measurement

- Hardware
 - z14 3906 model M04
 - LPAR with up to 32 dedicated CPs + 3 zIIPs
 - Separate LPAR for network driver with up to 8 dedicated CPs + 3 zIIPs
 - DASD DS8870
 - Internal Coupling Facility with ICP links
- Software
 - z/OS V2.3
 - Db2 V11
 - CICS TS V5.4 refresh November 2017
 - CICS TS V5.5 GA level

Notes

This is the environment we used to test the final build of the CICS TS V5.5 release.

Release to release comparisons (V5.5)

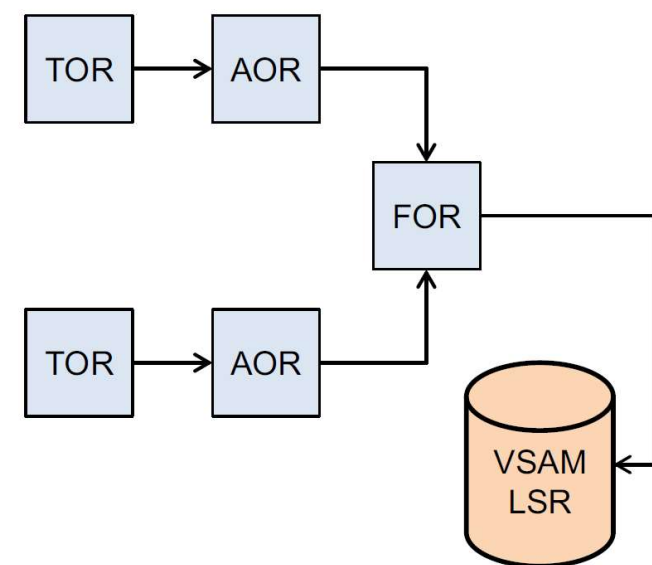
Notes

Many workloads are executed to ensure the performance of CICS is not degraded when upgrading from one release to another. In this case comparing v5.4 to v5.5.

It is very important that customers do not experience a reduction in performance, or an increase in CPU when upgrading to the latest release.

DSW static routing workload overview

- COBOL/VSAM
- Average of 6 file control requests per transaction
 - 69% Read, 10% Read for Update, 9% Update, 11% Add, 1% Delete
- All transactions statically routed from 2 TORs to 2 AORs
- File control requests function-shipped to 1 FOR
- Non-threadsafe workload



Notes

DSW – data system workload

A very old benchmark, but still represents a great number of applications found in production workloads.

The application only executes on the QR TCB and has minimal business logic, but exercises some of the core CICS functionality.

DSW workload V5.4 vs V5.5

ETR	CICS %	ms/tran
4180.81	76.37%	0.183
4938.15	89.21%	0.181
6054.32	108.09%	0.179
6582.22	116.78%	0.177
7143.83	126.76%	0.177

CICS TS V5.4
 Average CPU / tran = 0.179ms

ETR	CICS %	ms/tran
4180.34	73.22%	0.175
4948.82	85.95%	0.174
6057.17	103.90%	0.172
6591.39	112.31%	0.170
7151.20	121.70%	0.170

CICS TS V5.5
 Average CPU / tran = 0.172ms

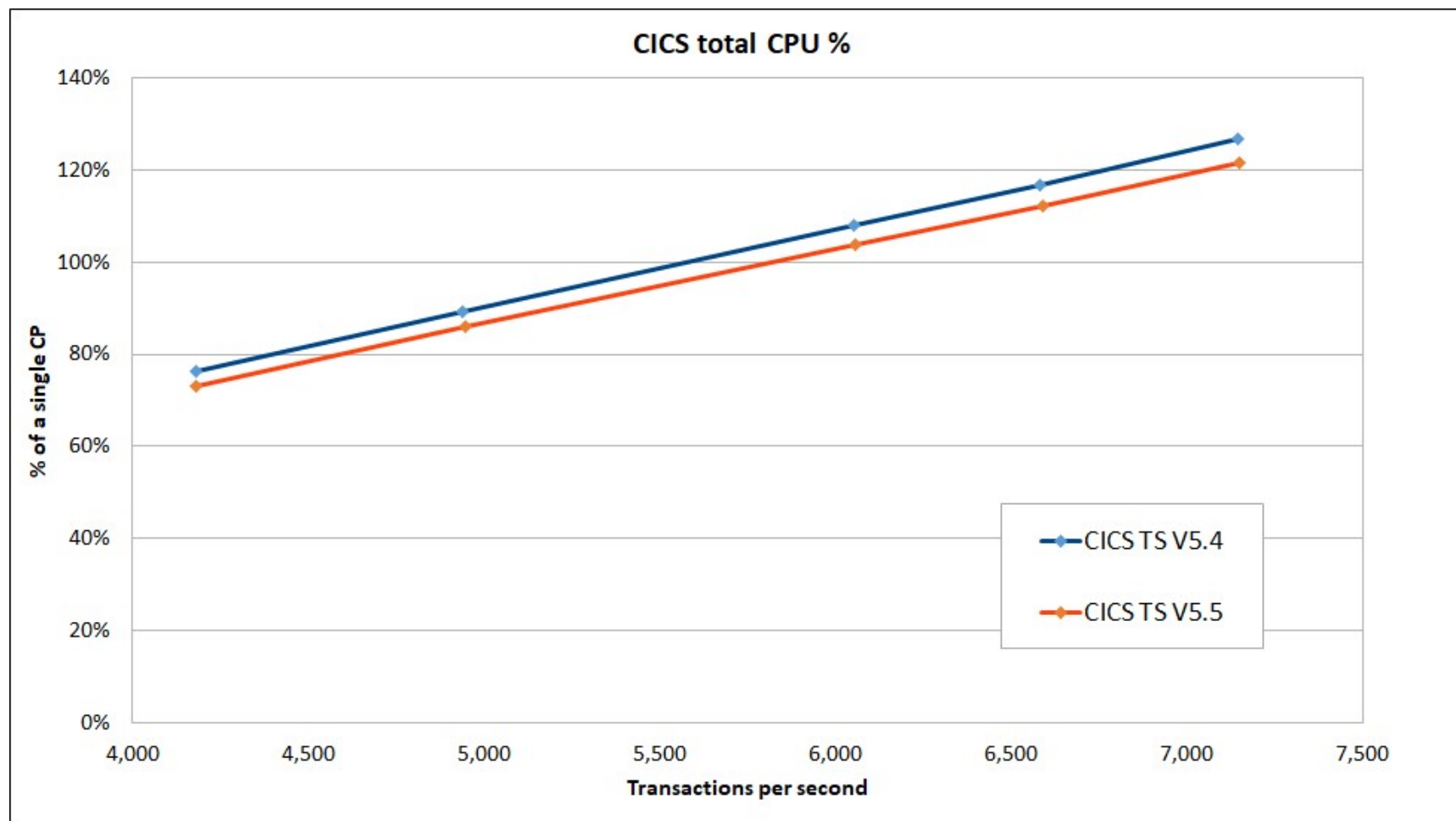
Notes

The table shows RMF data extracted from 5 different transactions rates for both CICS TS V5.4 and CICS TS V5.5.

A small performance improvement when averaged over the 5 intervals (7 μ s or 4%), however it's unlikely to translate into a major saving in a production environment.

Net takeaway is that V5.5 shows no performance regression when compared to V5.4 for this workload.

DSW workload V5.4 to V5.5



Notes

The same data, but presented as a line chart.

The straight line indicates linear CPU usage as the workload increases, with V5.5 being slightly cheaper per transaction than V5.4 as we saw in the raw data.

RTW workload overview

- COBOL/DB2 threadsafe application
- 7 transaction types
- 20 database tables
- Average 200 DB2 calls per transaction
 - 54% select, 1% insert, 1% update, 1% delete
 - 8% open cursor, 27% fetch cursor, 8% close cursor

Notes

RTW – relational transactional workload

This is a threadsafe application which exercises the CICS-Db2 interface.

This workload is more heavyweight than the DSW (COBOL-VSAM) workload we saw earlier.

RTW workload V5.4 to V5.5

ETR	CICS %	ms/tran
713.33	89.25%	1.251
996.88	124.43%	1.248
1417.03	177.47%	1.252
1959.66	248.73%	1.269
2401.43	309.99%	1.291

CICS TS V5.4
 Average CPU / tran = 1.262ms

ETR	CICS %	ms/tran
713.41	88.59%	1.242
997.00	123.74%	1.241
1417.54	176.81%	1.247
1960.32	248.39%	1.267
2402.72	309.49%	1.288

CICS TS V5.5
 Average CPU / tran = 1.257ms

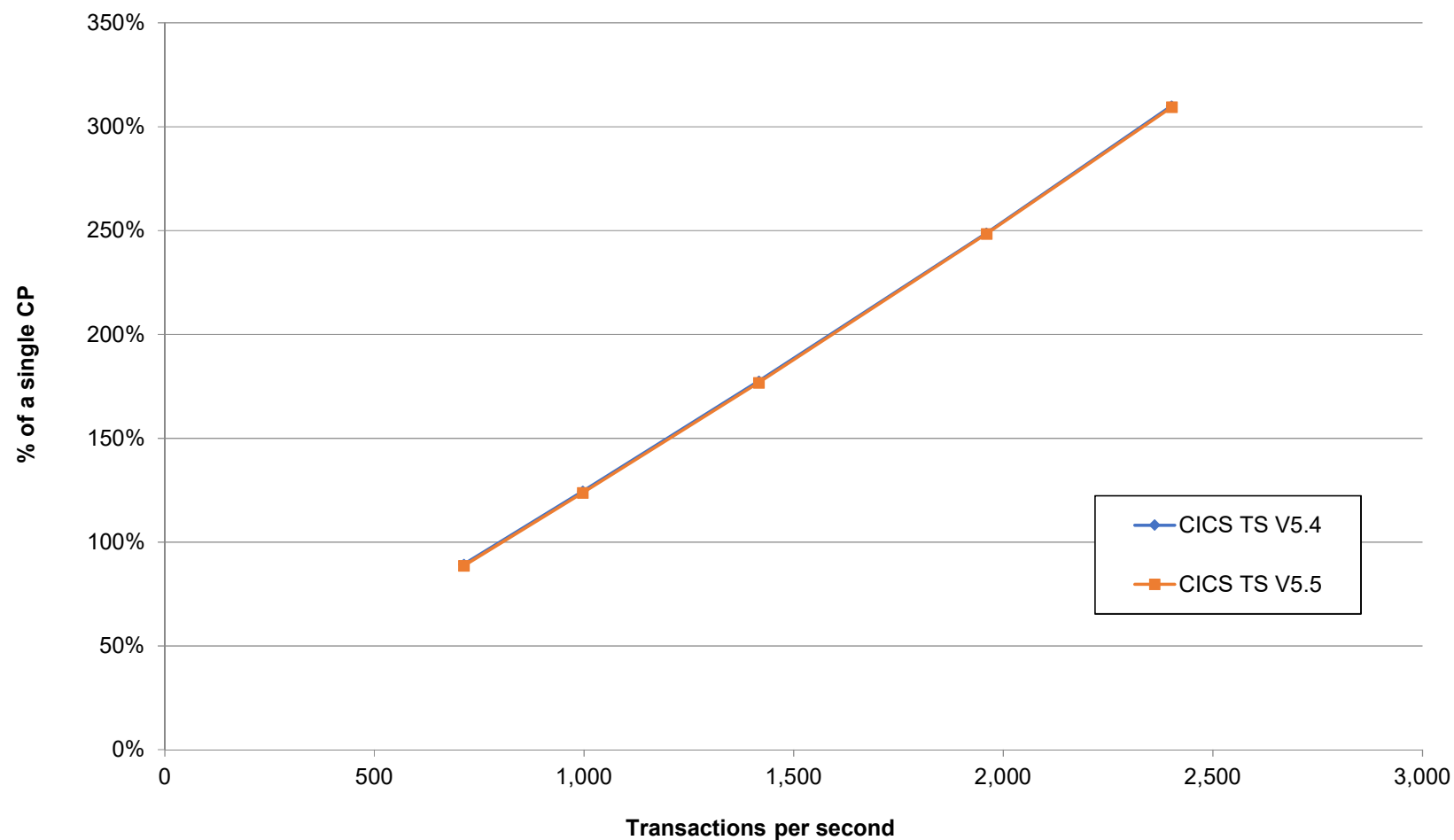
Notes

The table shows RMF data extracted from 5 different transactions rates for both CICS TS V5.4 and CICS TS V5.5.

Again, there is a slight performance improvement on a per-transaction basis. This is calculated as approximately the same gain as the DSW workload – here it is 5 μ s, but the relative size of that improvement is very small (less than 0.5%).

Net takeaway is that CICS TS V5.5 does not show performance degradation when compared to V5.4 for this workload.

RTW workload V5.4 to V5.5



Notes

The CICS TS V5.4 and V5.5 plots appear as a single line as the performance is very close.

Java performance

Notes

This section looks at how to improve performance by upgrading the JVM runtime used by CICS for Java applications.

Java 8 required for CICS TS V5.5

IBM 64-bit SDK for z/OS, Java Technology Edition	CICS TS						Liberty <=19.0.0.2	19.0.0.3+	Comments
	V5.1	V5.2	V5.3	V5.4	V5.5	V5.6 open beta			
V7.0	✓	✓	✓	✓			✓		<u>Out of service 30 Sep 2019</u>
V7.1	✓	✓	✓	✓			✓		Supported until 2022
V8.0	✓	✓	✓	✓	✓	✓	✓	✓	Supported until at least 2025

[Java 8 recommended for CICS TS V5](#) all releases

Also see [IBM FAQ to Oracle's Java Products Commercial Licensing](#)

Notes

With the correct PTFs, all releases of CICS TS V5 support Java V8.

Java hardware exploitation

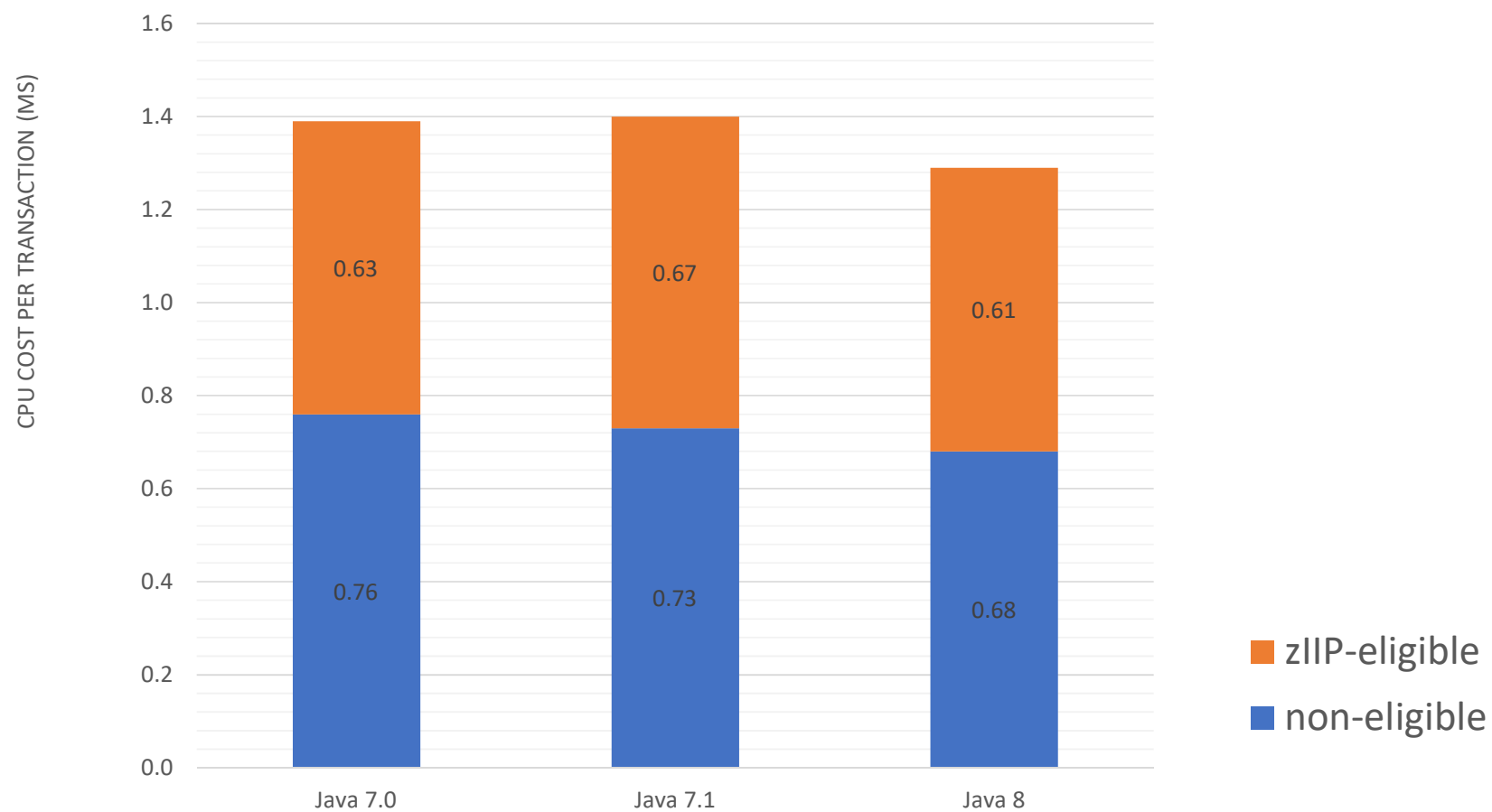
- Java 7.0 – z196 and zEC12 exploitation
 - New instructions
 - Transactional execution
- Java 7.1 – zEC12 exploitation
 - zEDC for zip acceleration
 - Increased zIIP offload from SR3 onwards
- Java 8.0 – language improvements and z13 exploitation
 - SIMD instructions
 - SMT mode 2
 - Use of crypto acceleration
- Java 8.0 SR5 – improved garbage collection and z14 exploitation

Notes

Each new release of the IBM Z hardware has an associated deliverable of the JVM for z/OS.

The latest release of the JVM can exploit the newer instructions found in the latest hardware, without an application change or even a recompilation of your Java applications.

Comparison of CPU for Java 8 vs Java 7



Notes

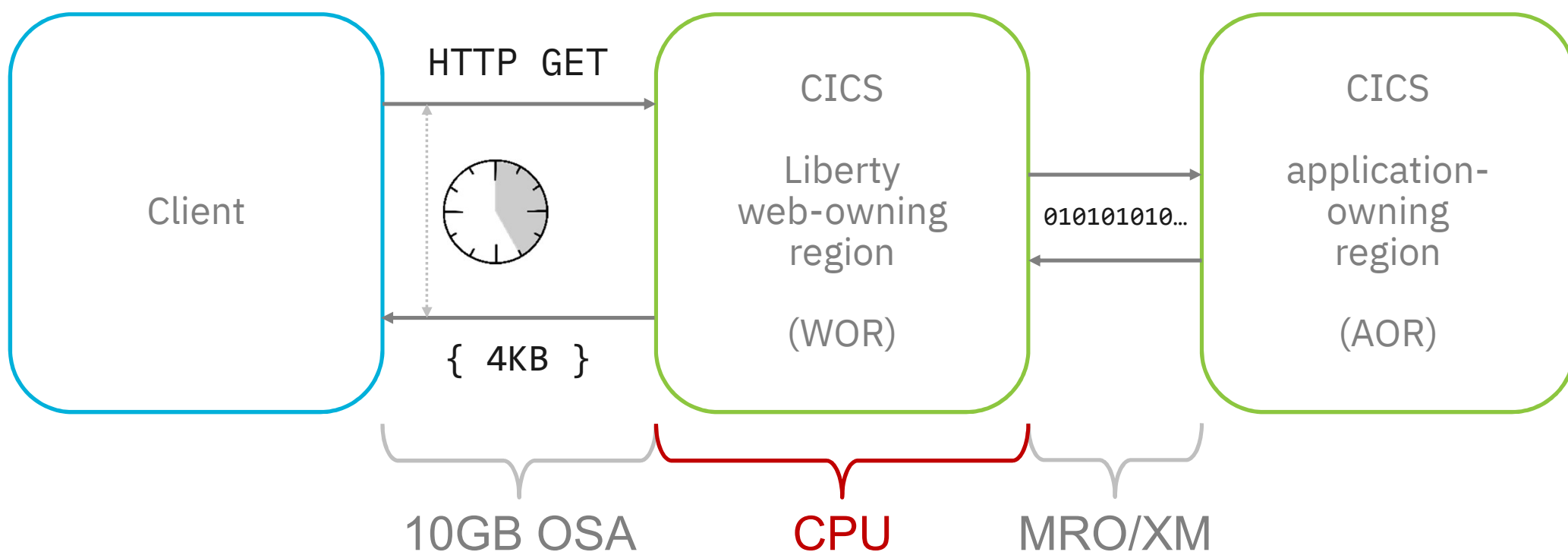
Chart uses an OSGi Java workload.

Several applications provide a mixture of operations, including JDBC access, VSAM access, string manipulation, and mathematical operations.

Overall CPU consumption of the Java 7.0 and 7.1 workloads are nearly identical, however a slightly larger fraction of the Java 7.1 workload is eligible for offload to a specialty engine.

The Java 8 runtime shows a clear overall reduction in CPU consumed, plus a reduction in cycles consumed on general-purpose processors.

SSL benchmark application configuration



Notes

The client performs an HTTP GET request for a resource in the Liberty web-owning region (WOR).

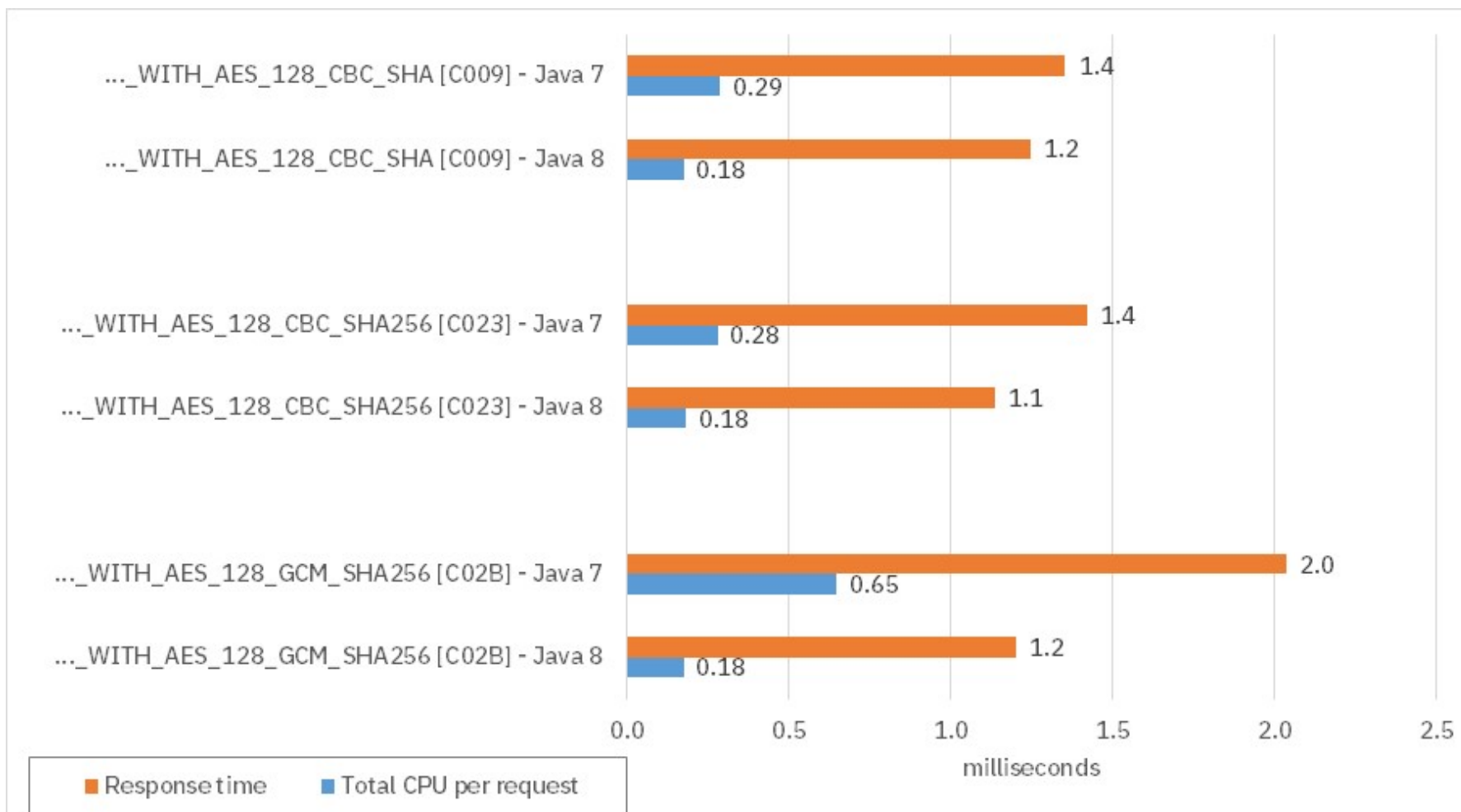
The WOR is running a simple JAX-RS application which then issues a link using the JCICS command to a program defined as remote in an application-owning region (AOR), connected using MRO/XM.

The application returns binary data to the WOR, which is then converted to 4KB of JSON data, and returned to the client.

The response time is measured at the client, and the CPU consumed is measured only in the WOR.

By using this configuration, we can split out the cost of the business logic from the cost of the data encryption.

SSL and Java



Notes

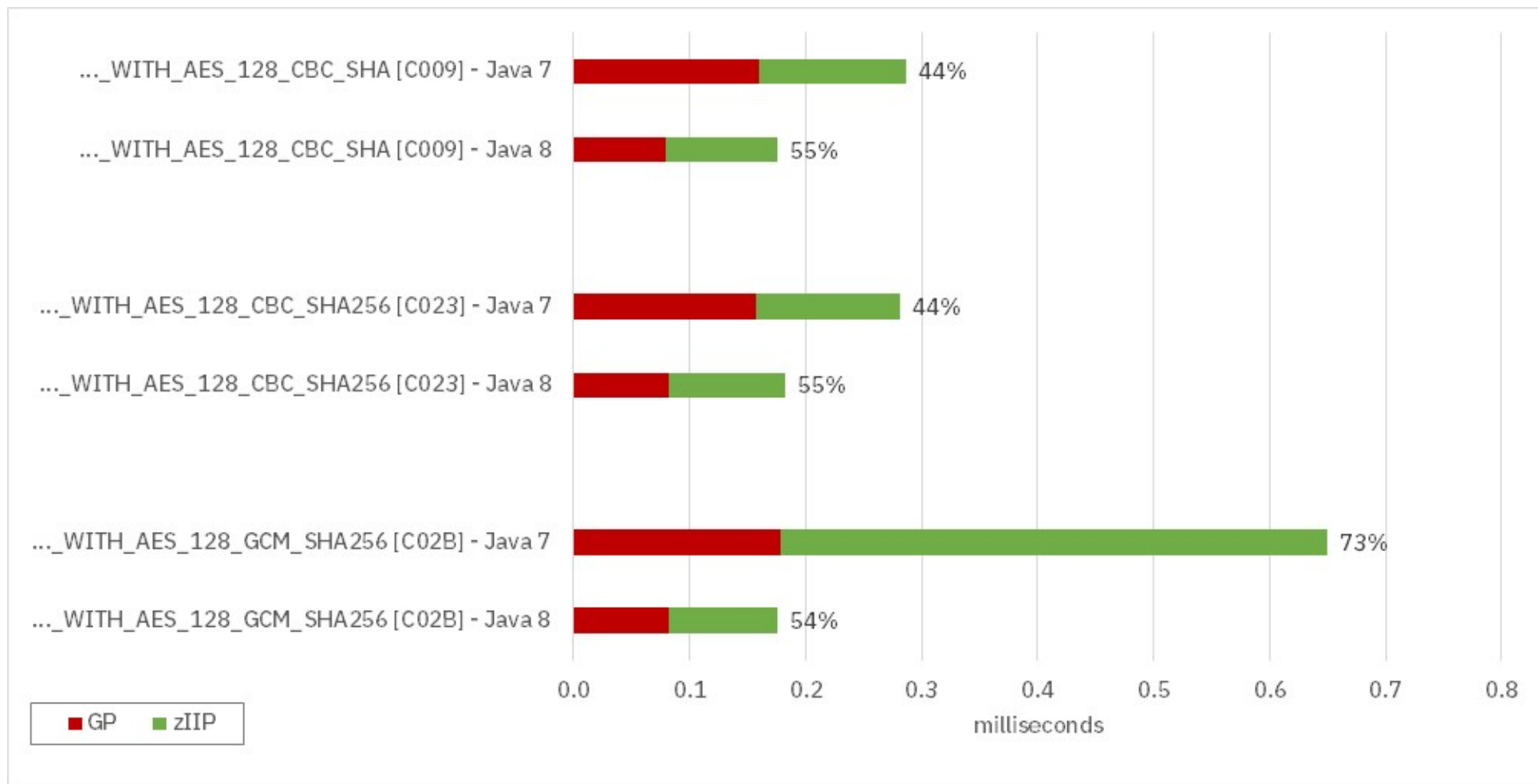
Each pair of horizontal bars represent the response time (orange) and CPU time (blue) for a given cipher suite with a specified level of the Java runtime.

For this case, the workload used persistent sessions therefore the first component of the cipher suite is irrelevant – only the block cipher and the message digest is shown. For completeness, the cipher suite number is also shown.

For the C009 and C023 ciphers, Java 8 shows a CPU and response time improvement – this is due to the Java runtime being able to take advantage of z14 hardware instructions that the Java 7 runtime could not, along with other JRE performance improvements.

For the C02B cipher, there is a significant performance improvement – this is because Java 8 SR5 can use the new KMA instruction that specifically assists with cipher suites which use GCM.

SSL and zIIP eligibility



Notes

This chart uses the same cipher suite and Java runtime level combinations as previously, but this time looks purely at the CPU consumption.

Red is general-purpose CPU, while green is zIIP-eligible CPU.

For the C009 and C023 ciphers, not only is the CPU reduced overall, the fraction which is eligible for offload has increased (44% to 55%).

The C02B cipher is the one which uses the KMA instruction for GCM processing. In this situation, the CPU consumed has decreased significantly with the introduction of this new instruction. The zIIP-eligible fraction is reduced here (73% to 54%): this is because what was previously implemented in Java can now be handed to the hardware. There's simply less Java code being executed.

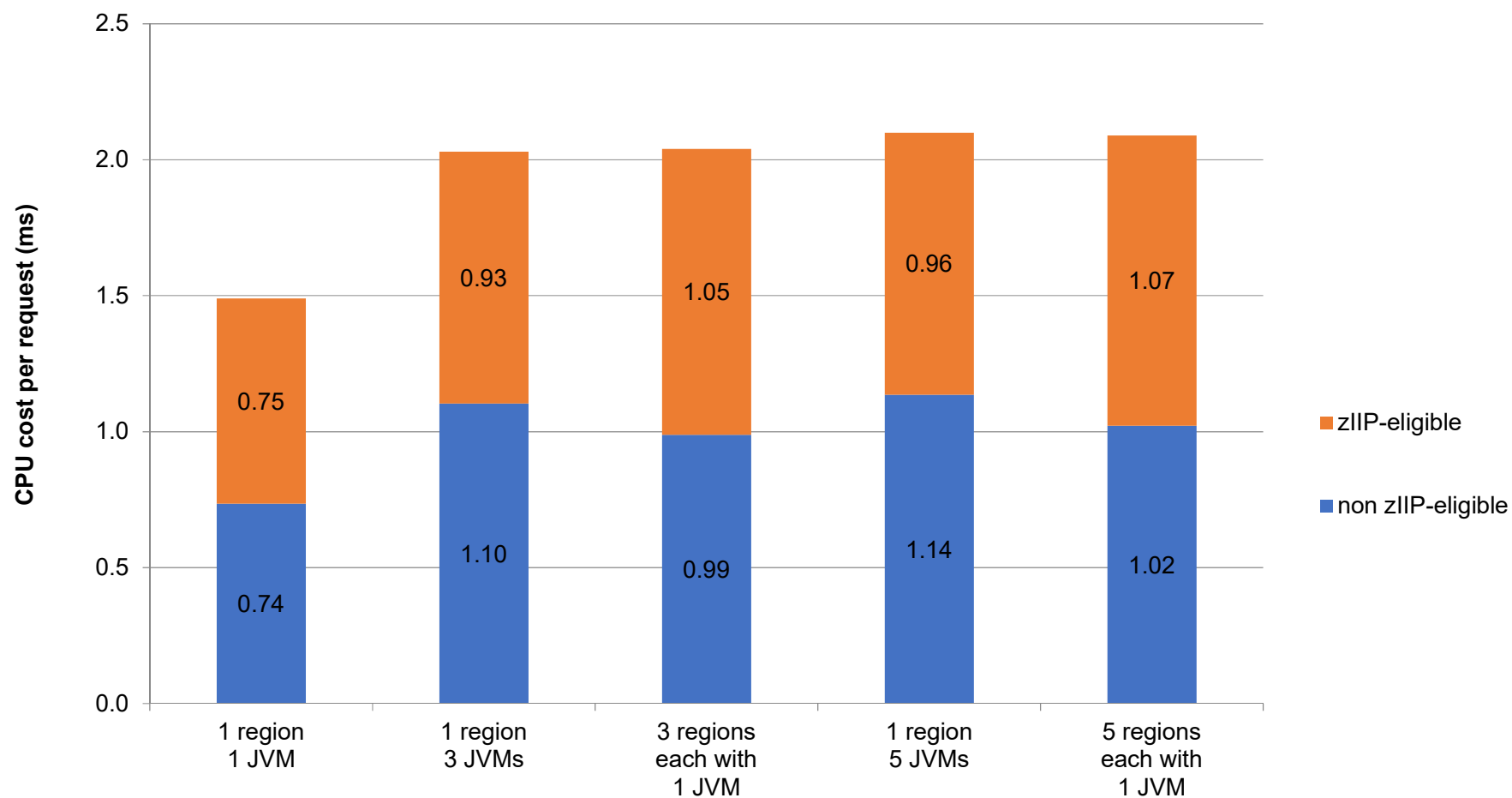
Multiple secure Liberty JVM servers

- CICS TS V5.5 enables multiple secured Liberty JVM servers in a single CICS region.
 - CICS TS V5.4 with APAR PI98174
- Our environment:
 - z/OS v2.3, Java 8.0 SR5
 - CICS TS V5.5 in development version
 - RMODE64 enabled
 - Compressed reference enabled.
 - Shared libraries disabled.

Notes

Disabling shared libraries so that any address space won't reserves an area of 31-bit virtual storage that is equal in size to the value of the z/OS SHRLIBRGNSIZE parameter, which is likely to increase the virtual storage footprint of each region.

Multiple Liberty performance - CPU



Notes

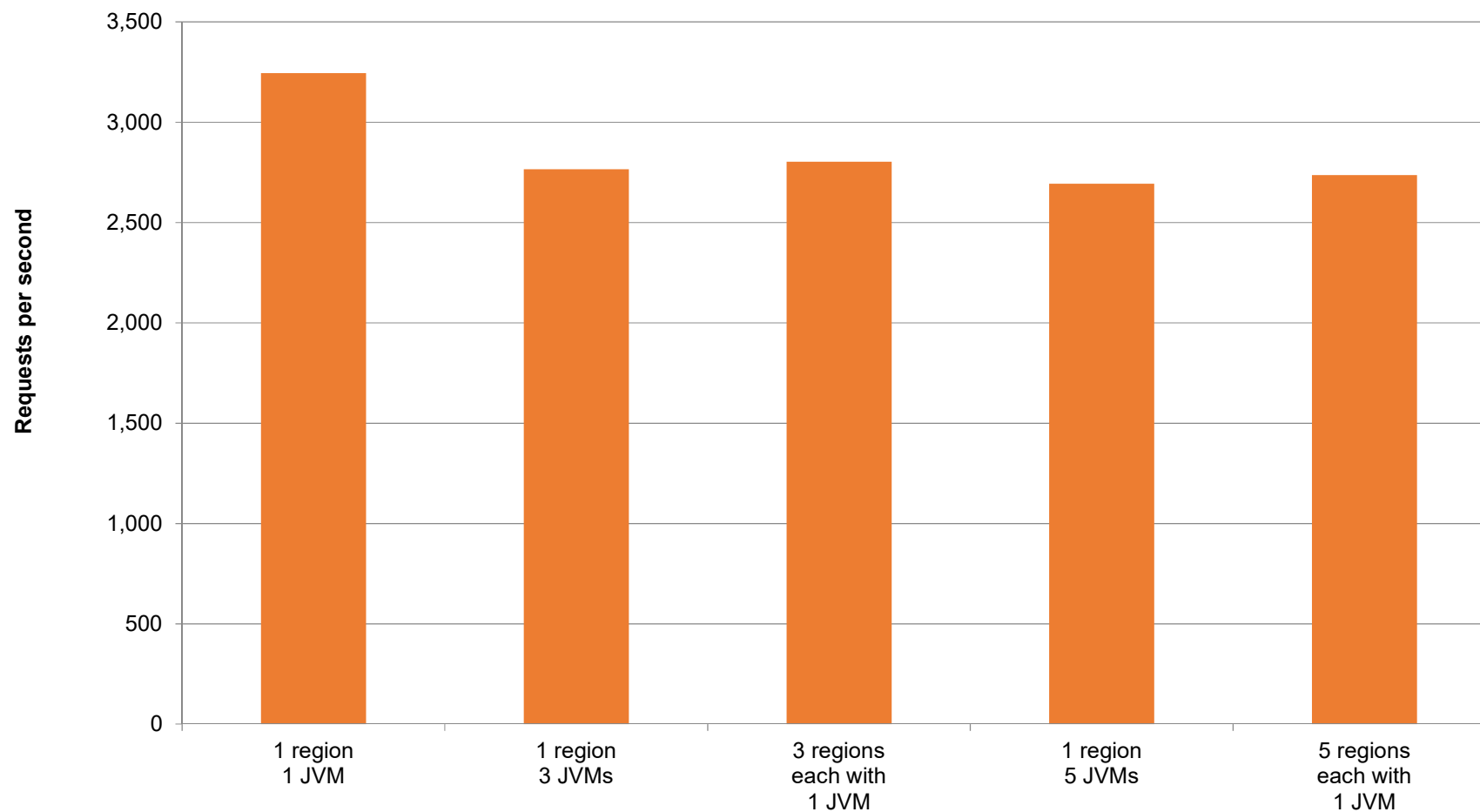
The lowest CPU cost per request was provided by single JVM server in a single CICS region (configuration 1).

This is because the JVM in configuration 1 will process more requests than each individual JVM used in configurations 2 through 5. The more requests that are processed by a JVM, the more effectively the JIT compiler can optimize the code path, resulting in a lower CPU per request.

configuration 1 (one CICS region with one Liberty JVM server) had optimized a total of 7,079 Java methods.

Conversely, one of the JVMs in configuration 5 (five CICS regions each with one Liberty JVM server) had optimized only 1,362 Java methods.

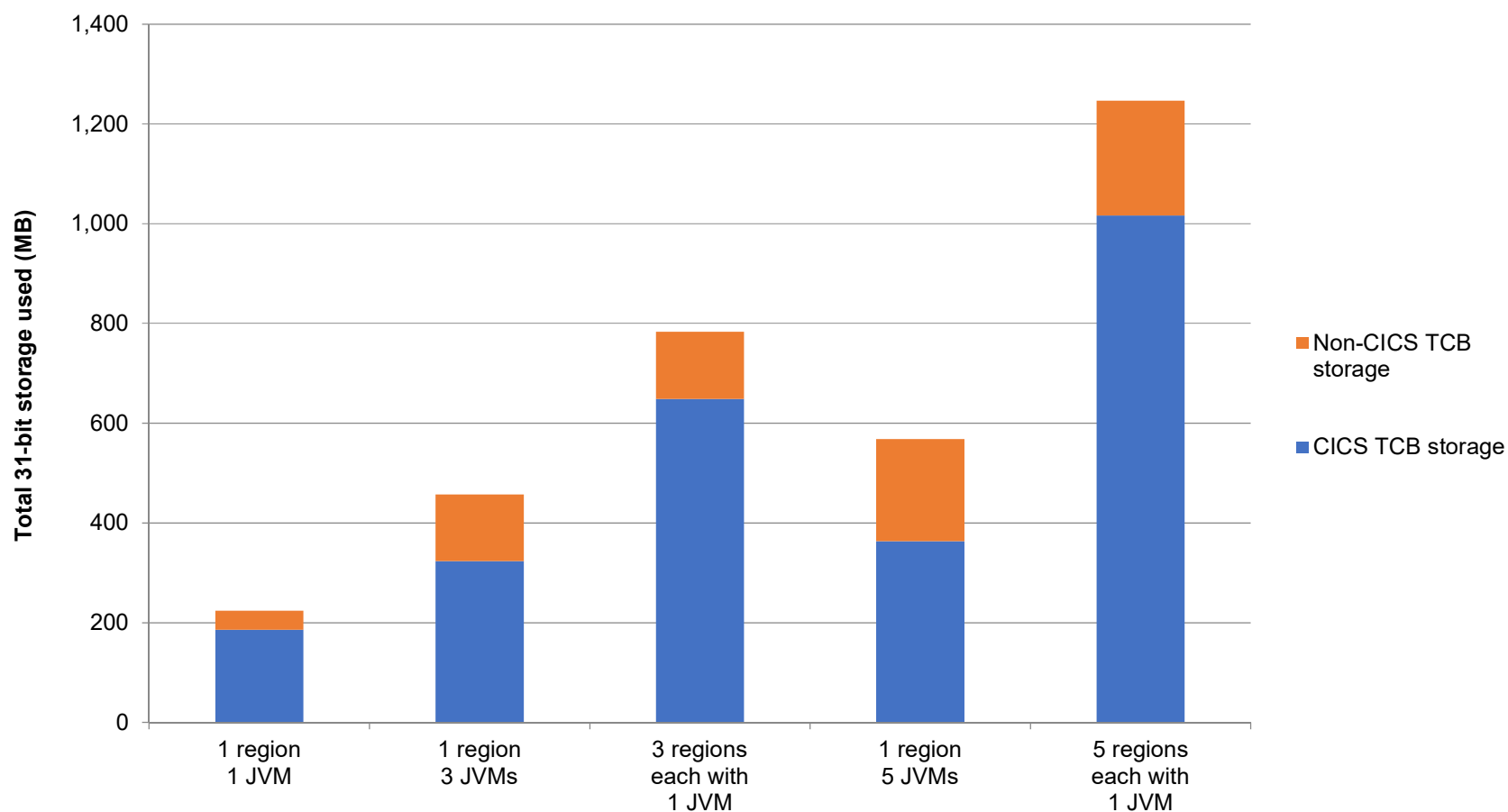
Multiple Liberty performance - throughput



Notes

This chart shows the throughput of the configurations, from client side point of view. As you can see, having multiple liberty server in one region doesn't increase the throughput. However you can get the isolation of java app and increase application availability by running them on multiple liberty servers in one region.

Multiple Liberty performance – 31-bit storage

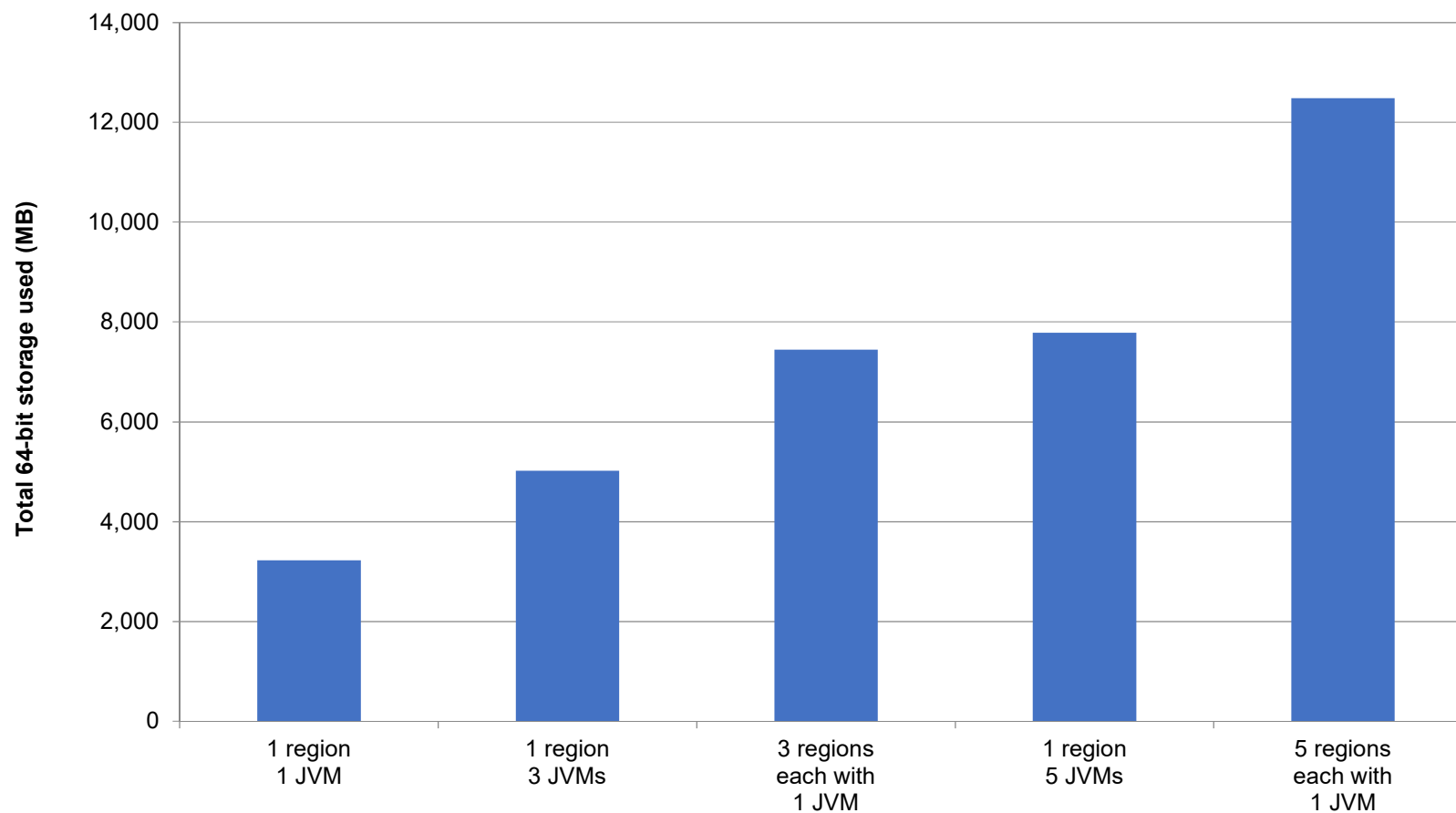


Notes

The amount of CICS TCB storage per CICS region is related to the number of concurrent tasks and TCBs used. To restrict the number of concurrent TCBs in a CICS region for a Java workload, use the THREADLIMIT attribute of the JVMSERVER resource definition.

The amount of non-CICS TCB storage that is used per CICS region is related to the number of JVM servers.

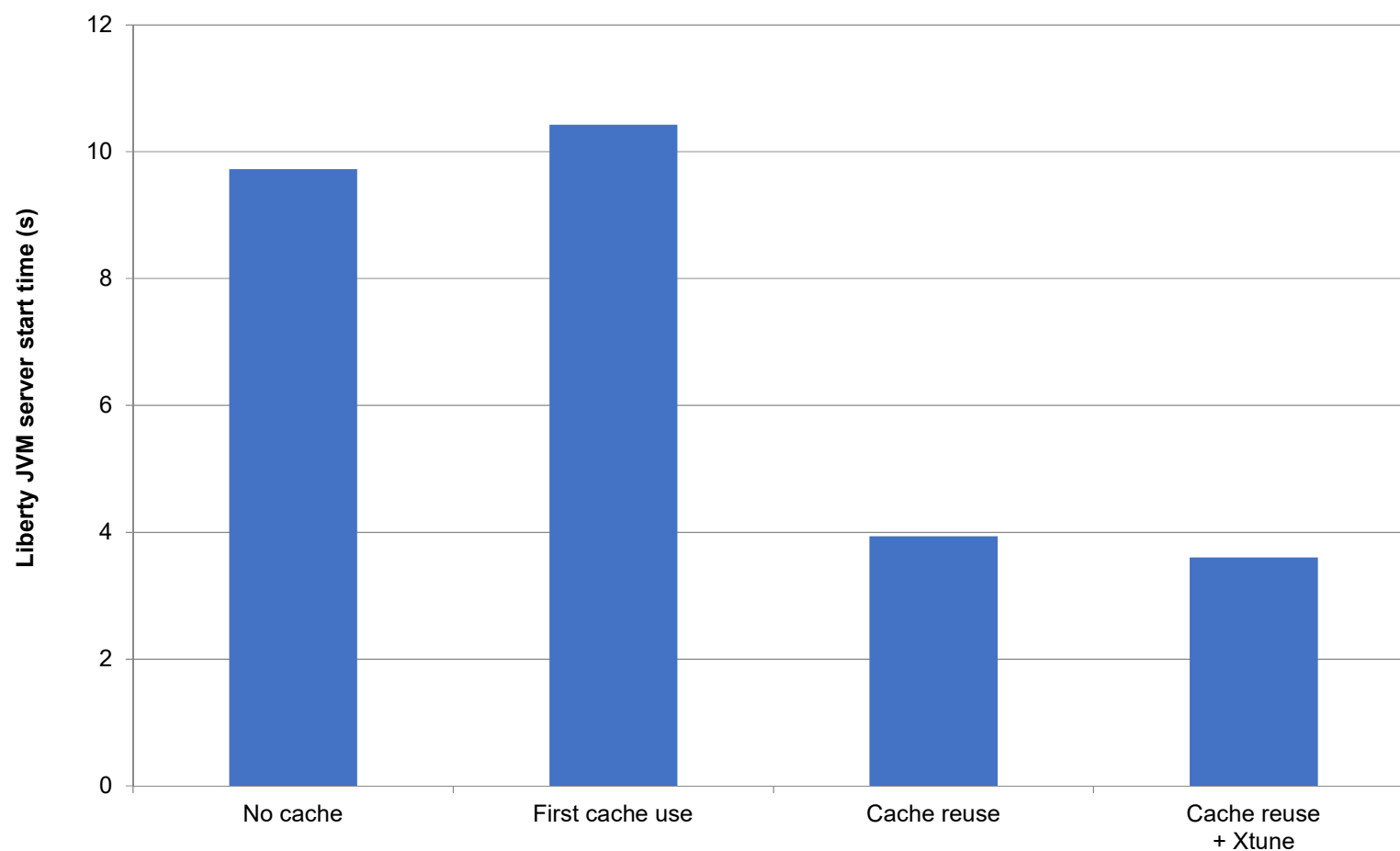
Multiple Liberty performance – 64-bit storage



Notes

Multiple JVMs in a single CICS region gives a reduction in overall 64-bit storage used. This is similar to 31-bit of storage usage.

Liberty JVM server start-up time -Xshareclasses



Notes

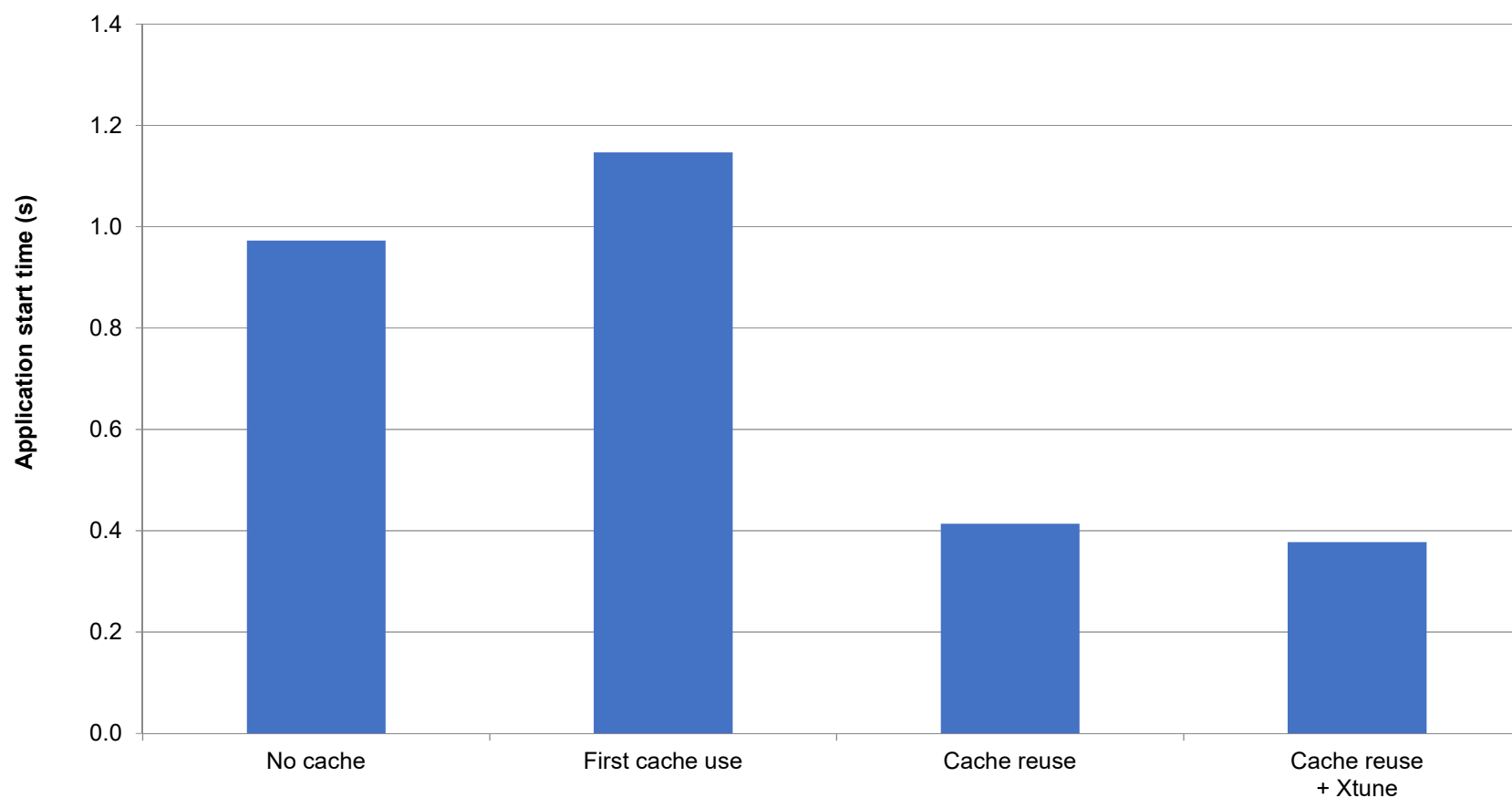
The class sharing feature (<https://ibm.biz/BdzSkf>) offers the transparent and dynamic sharing of data between multiple JVMs.

When enabled, JVMs use shared memory to obtain and store data, including information about: loaded classes, Ahead-Of-Time (AOT) compiled code, commonly used UTF-8 strings, and Java Archive (JAR) file indexes.

The first use of a shared class cache slightly increases startup times for both the Liberty JVM server and any applications. However, subsequent starts are significantly improved with shared class cache enabled.

The use of the `-Xtune:virtualized` JVM option further improves JVM and application startup time (<https://ibm.biz/BdzSkv>).

Liberty application start-up time -Xshareclasses



Notes

None

Threadsafe improvements

Notes

To reduce TCB switching and contention on the QR TCB, the CICS development team are always looking for opportunities to make CICS API and SPI commands threadsafe.

This section covers the reasons why this is important, along with a summary of recent changes in the V5 releases.

Program concurrency recap

- We run CICS with STGPROT=YES
- My application ...
 - ... runs USER key
 - ... is threadsafe
 - ... makes DB2 calls
- How do I maximize time spent on an Open TCB?

Notes

A recap of the CONCURRENCY attribute for a PROGRAM resource.

CICS TS V4.1 TCB Switching

STGPROT	Exec key	CONCURRENCY	API	Initial TCB	DB2 or MQ command	Threadsafe command	Non-threadsafe command
Yes/No	(any)	QUASIRENT	CICS	QR	QR → L8 → QR	no change	no change
		THREADSAFE		QR	L8	no change	QR
No	(any)	THREADSAFE	OPEN	L8	no change	no change	L8 → QR → L8
		THREADSAFE		L8	no change	no change	L8 → QR → L8
Yes	CICS	THREADSAFE	OPEN	L8	no change	no change	L8 → QR → L8
Yes	USER	THREADSAFE	OPEN	L9	L9 → L8 → L9	no change	L9 → QR → L9
		THREADSAFE		L9	L9 → L8 → L9	no change	L9 → QR → L9

Notes

A table showing the combinatorial effects of the STGPROT, EXECKEY, CONCURRENCY and API configuration options. This table is also presented in the CICS TS V5 Performance Report, IBM Redbooks publication SG24-8298.

<https://www.redbooks.ibm.com/abstracts/sg248298.html?Open>

CICS TS V4.2+ TCB Switching

STGPROT	Exec key	CONCURRENCY	API	Initial TCB	DB2 or MQ command	Threadsafe command	Non-threadsafe command
Yes/No	(any)	QUASIRENT	CICS	QR	QR → L8 → QR	no change	no change
		THREADSAFE		QR	L8	no change	QR
		REQUIRED		L8	no change	no change	L8 → QR → L8
No	(any)	THREADSAFE	OPEN	L8	no change	no change	L8 → QR → L8
		REQUIRED					
Yes	CICS	THREADSAFE	OPEN	L8	no change	no change	L8 → QR → L8
		REQUIRED					
Yes	USER	THREADSAFE	OPEN	L9	L9 → L8 → L9	no change	L9 → QR → L9
		REQUIRED					

Notes

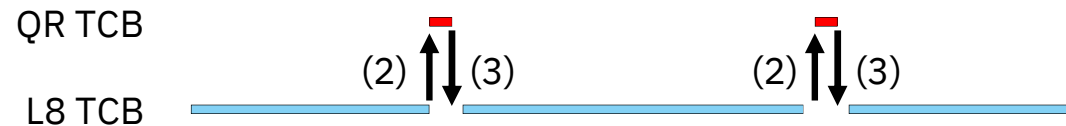
This page updates the table to include the CONCURRENCY(REQUIRED) option added in CICS TS V4.2.

Threadsafe Transient Data

V4.1 – CONCURRENCY(THREADSAFE)



V4.2 – CONCURRENCY(REQUIRED)



V5.1 – CONCURRENCY(REQUIRED) with threadsafe TD



- (1) TCB switch due to Db2 call
- (2) TCB switch due to EXEC CICS WRITEQ TD command
- (3) TCB switch back to L8 due to CONCURRENCY(REQUIRED)

Notes

Application is defined as API(CICSAPI) to avoid the use of API(OPENAPI).

Chart shows an application which alternately executes DB2 SQL calls and then WRITEQ TD commands.

Threadsafe Transient Data

Tran	#Tasks	Avg Response Time	Avg User CPU Time	Avg QR CPU Time	Avg KY8 CPU Time	Avg DSCHMDLY Count	TD	Avg Total Count	RMI	Avg DB2 Time
TDQ1	5938	.011942	.006967	.004597	.002370	302		150		.001626

V4.1
 QR = 4.60ms
 L8 = 2.37ms
 302 TCB switches

Tran	#Tasks	Avg Response Time	Avg User CPU Time	Avg QR CPU Time	Avg KY8 CPU Time	Avg DSCHMDLY Count	TD	Avg Total Count	RMI	Avg DB2 Time
TDQ1	5992	.011393	.006875	.000212	.006663	306		150		.001420

V4.2
 QR = 0.21ms
 L8 = 6.66ms
 306 TCB switches

Tran	#Tasks	Avg Response Time	Avg User CPU Time	Avg QR CPU Time	Avg KY8 CPU Time	Avg DSCHMDLY Count	TD	Avg Total Count	RMI	Avg DB2 Time
TDQ1	6000	.006805	.006195	.000026	.006169	8		150		.001147

V5.1
 QR = 0.03ms
 L8 = 6.17ms
 8 TCB switches

Notes

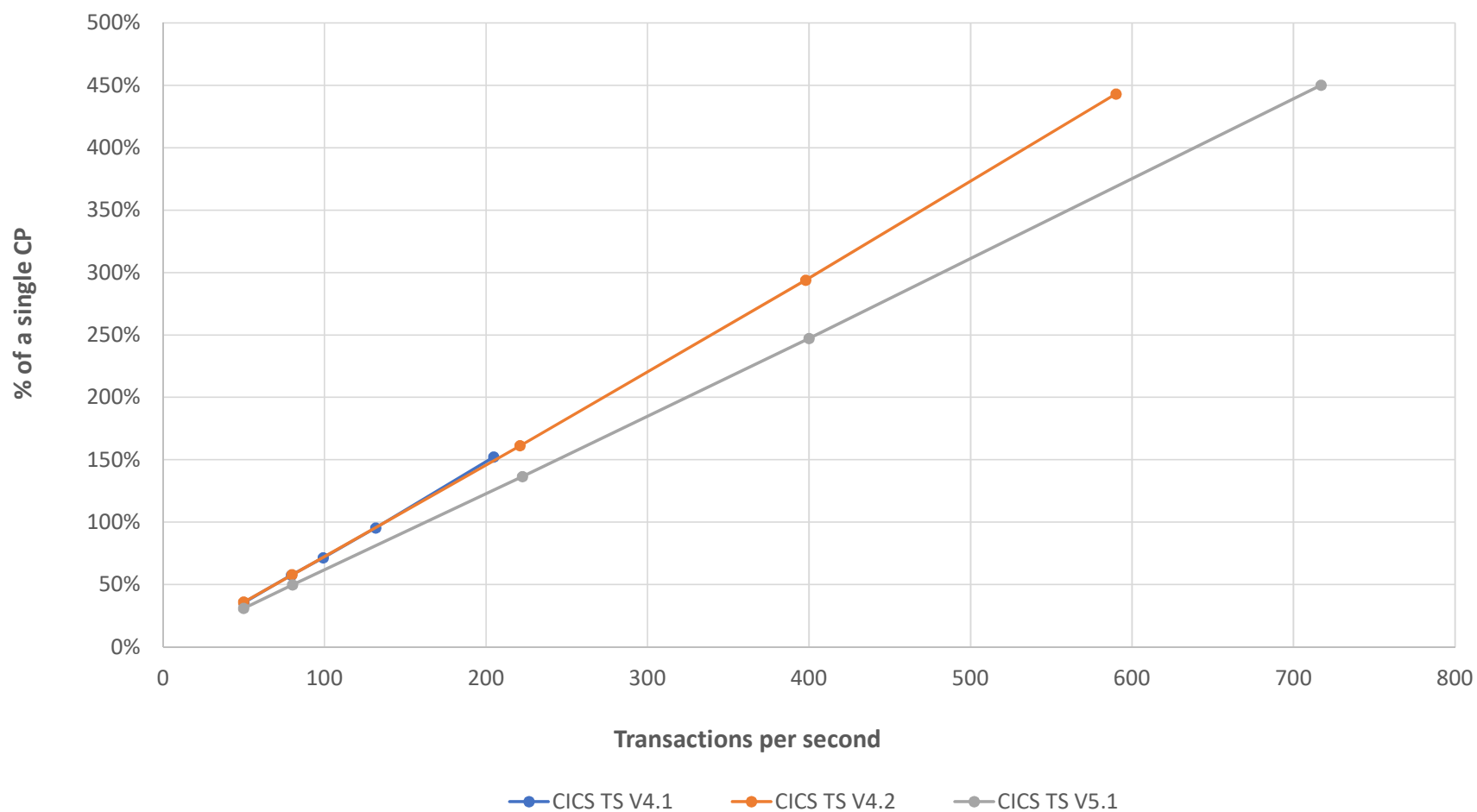
Chart shows an extracts from CICS Performance Analyzer reports for each of the various CICS TS levels.

V4.1 shows a significant number of TCB switches, with a large fraction of CPU consumed on the QR TCB.

V4.2 introduced CONCURRENCY(REQUIRED), which does not reduce the TCB switches, but reduces significantly the amount of CPU time the application spends executing on the QR TCB.

V5.1 introduces threadsafe transient data, which removes the need to switch the QR TCB for the WRITEQ TD command.

Transient Data mixed with Db2



Notes

Note that the V4.1 line hits a limit around the 210 transactions per second mark. This is because each transaction costs around 4.60ms of CPU time on the QR TCB. Therefore, the maximum throughput for this transaction will be:

$$1000 \text{ ms} / 4.60 \text{ ms/tran} = 217 \text{ transactions per second.}$$

The V4.2 and V5.1 lines do not see this limit as there is significantly less CPU time spent on the QR TCB.

The V5.1 line is slightly lower than the V4.2 line due to the reduction in CPU cost of the incurred TCB switches.

Threadsafe improvements (V5.5)

- Access to CFDTs is now threadsafe
 - No TCB switch for access
 - Syncpoint also can occur on Open TCB
 - Initial open and load occurs on QR TCB
- EXEC CICS QUERY SECURITY
 - Reduce number of TCB switches to RO
- With APAR PH05298 application running on an open TCB will not switch to CO TCB when it uses CICS auxiliary temporary storage, so save TCB switching.

Notes

Access to Coupling Facility data tables (CFDTs) is now threadsafe.

CFDTs can therefore be accessed by applications that run on open task control blocks (TCBs) without incurring a TCB switch. Syncpoint processing of CFDTs can also run on an open TCB. However, the opening and loading of a CFDT still occurs on a quasi-reentrant (QR) TCB.

In earlier releases, CICS required multiple TCB switches to complete the EXEC CICS QUERY SECURITY call, and these have been optimized to just a pair of switches to the RO TCB and back.

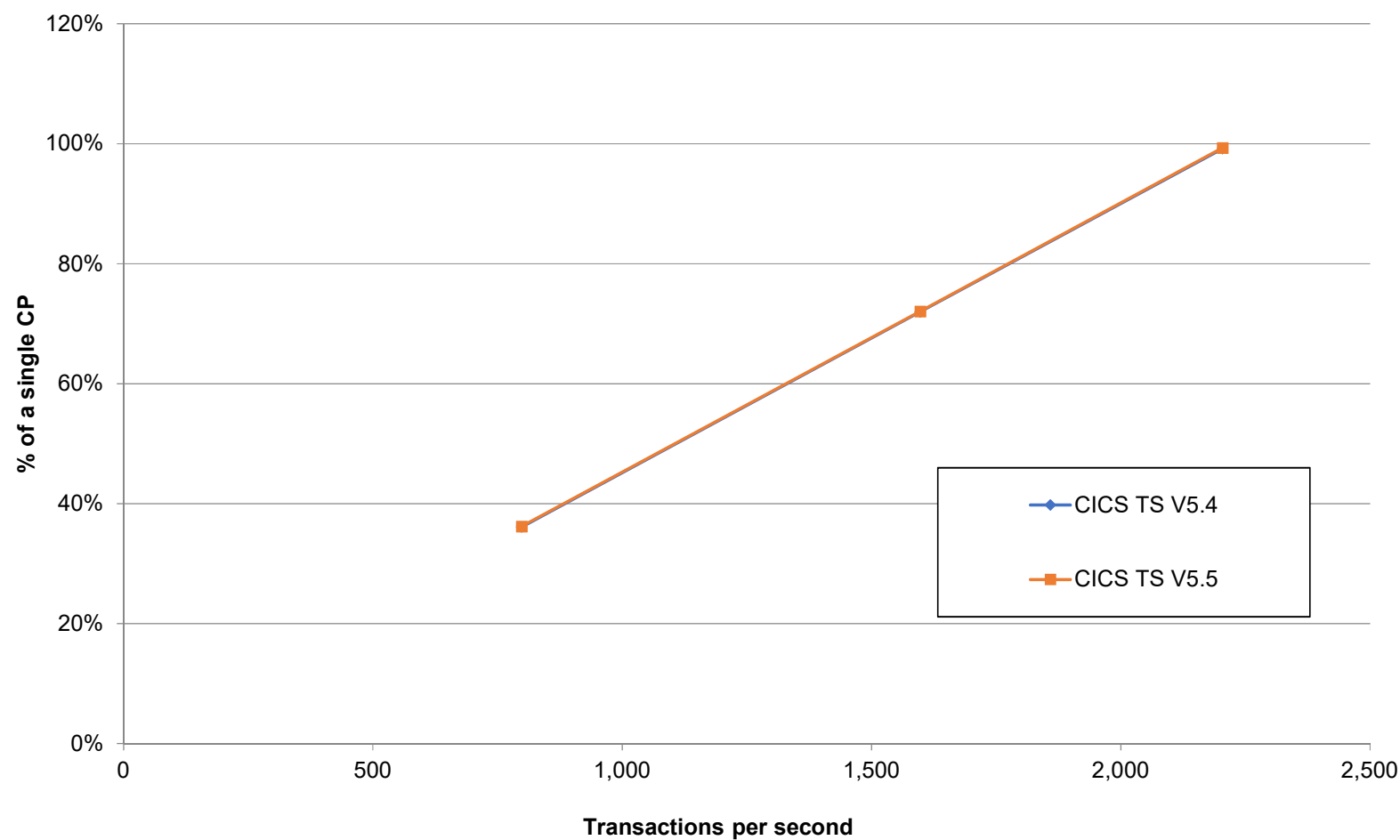
Coupling Facility Data Table workload

- Each transaction accesses 50 records
 - Read-only transactions – 1 access per record
 - EXEC CICS READ
 - Update transactions – 2 accesses per record
 - EXEC CICS READ UPDATE, EXEC CICS REWRITE
- Workload mix:
 - 70% read only transactions, 30% update transactions
- CF data table access mix (average 65 requests per transaction):
 - EXEC CICS READ – 35 (54%)
 - EXEC CICS READ UPDATE – 15 (23%)
 - EXEC CICS REWRITE – 15 (23%)
- All reads use SHA-256 hash to validate correctness of record

Notes

A high level description of the performance test workload.

CFDT non-threadsafe workload

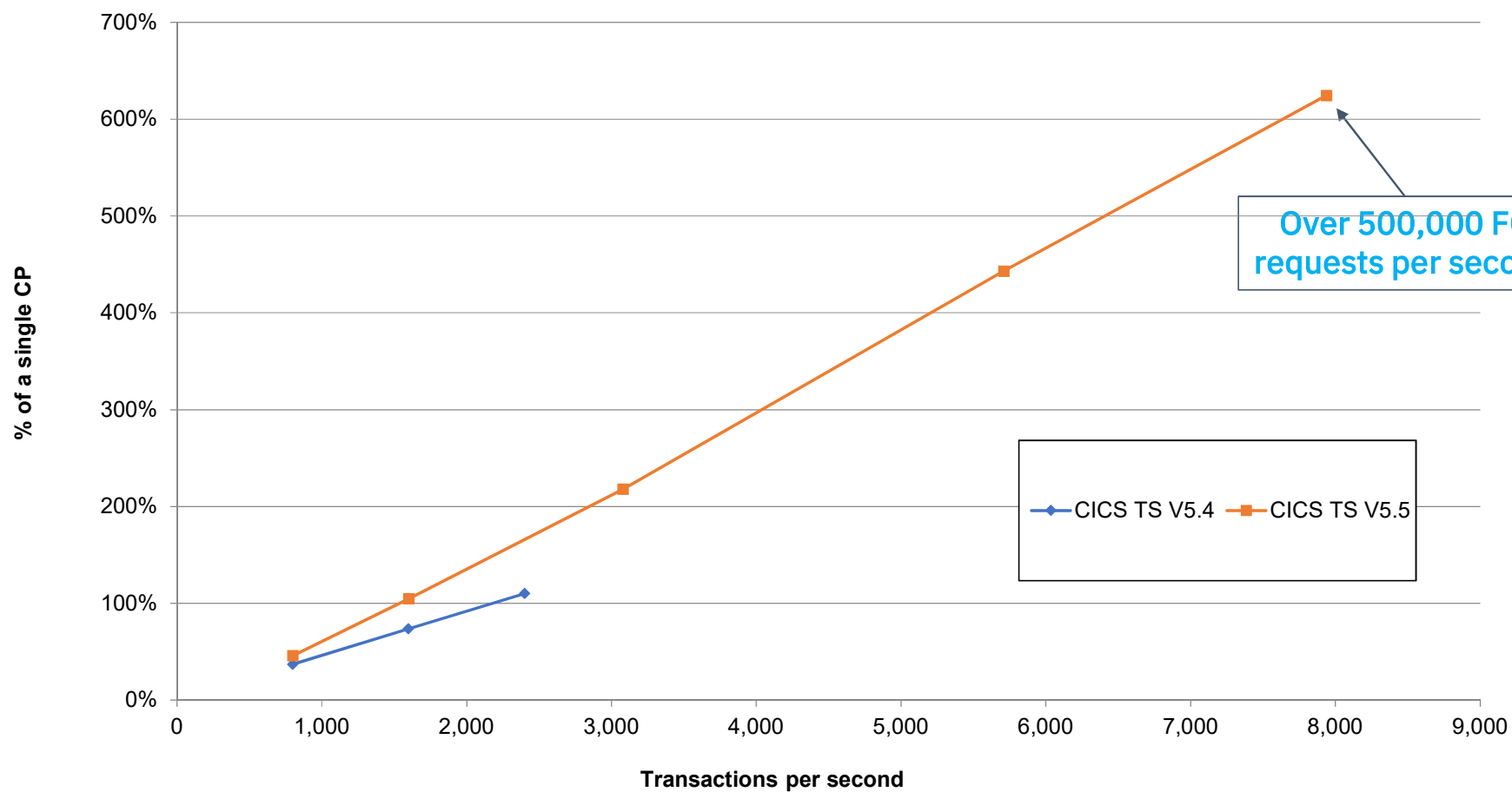


Notes

When making CFDT access threadsafe, we first need to verify that non-threadsafe applications do not suffer a performance overhead when upgrading to V5.5.

As shown on this chart, the performance for a non-threadsafe workload that uses coupling facility data tables is almost identical across the two releases.

CFDT threadsafe workload - CPU



Notes

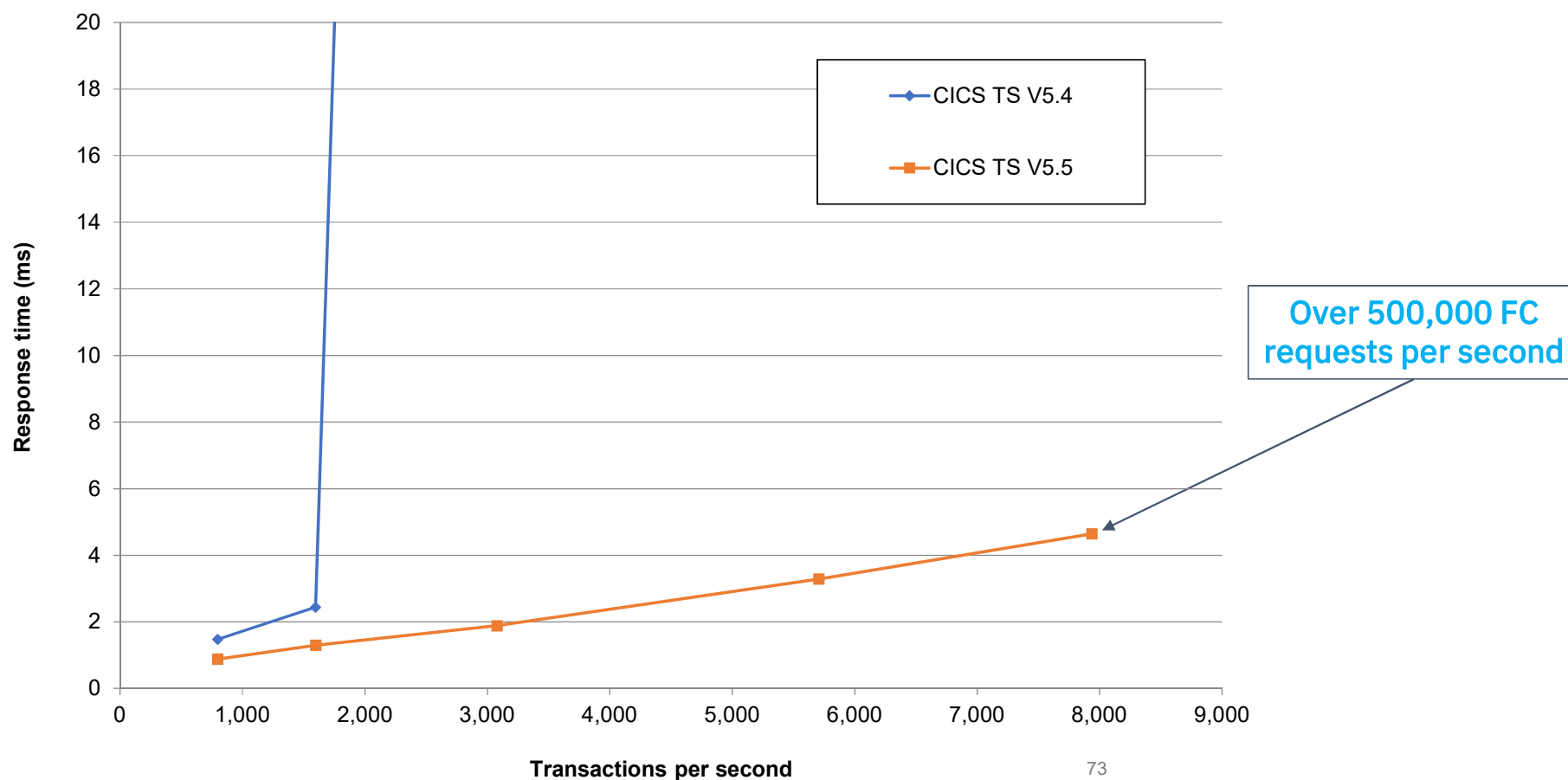
Having established that non-threadsafe access is not degraded, we can now measure how well coupling facility data table access scales. The workload is defined with CONCURRENCY (THREADSAFE) so it starts on L8 TCB at 5.4 and 5.5. V5.4 will immediately encounter a TCB switch to the QR TCB, but V5.5 will remain on the Open TCB throughout the application.

Clearly V5.5 shows much better scalability, but the V5.4 workload is limited by contention on the QR TCB.

Threadsafe access to the CFDT allows for much greater concurrency within CICS as each of the application tasks run on Open TCBs. This increase in concurrency will necessarily cause an increase in CPU cost per transaction due to the significantly increased number of concurrent TCBs executing within the z/OS LPAR. Therefore, the CPU per transaction results cannot be directly compared.

At the peak of 8,000 transactions per second a single CICS region is issuing over 500,000 file control requests per second. This workload alone drove our CF with 2 dedicated CPs to over 70% utilisation.

CFDT threadsafe workload – response time



Notes

CFDT response time chart shows that the V5.4 response time goes way off the chart because all requests are waiting for QR dispatch because the QR TCB is completely flat-out, while the V5.5 response time is not constrained by QR TCB time.

Notes

This shows the CICS dispatcher statistics from V5.4 and V5.5.

In the V5.4 report, the workload is constrained by the QR TCB as it is dispatched for over 99.9% of the 5 minute interval. The CICS region is at maximum capacity.

In the V5.5 report, the QR TCB is dispatched for less than 50% of the 5 minute interval.

Channels performance improvement

- CICS V5.5 improved performance for applications where many containers are stored in a single channel.
 - By using a hash table to access containers, rather than searching a list.
- This changes the order in which containers are returned when browsing a channel.
 - So the order in which containers are returned from EXEC CICS GETNEXT CONTAINER (CHANNEL) can be different
 - The CICS feature toggle `com.ibm.cics.container.hash` can be set to `false` to restore CICS to the previous behaviour.

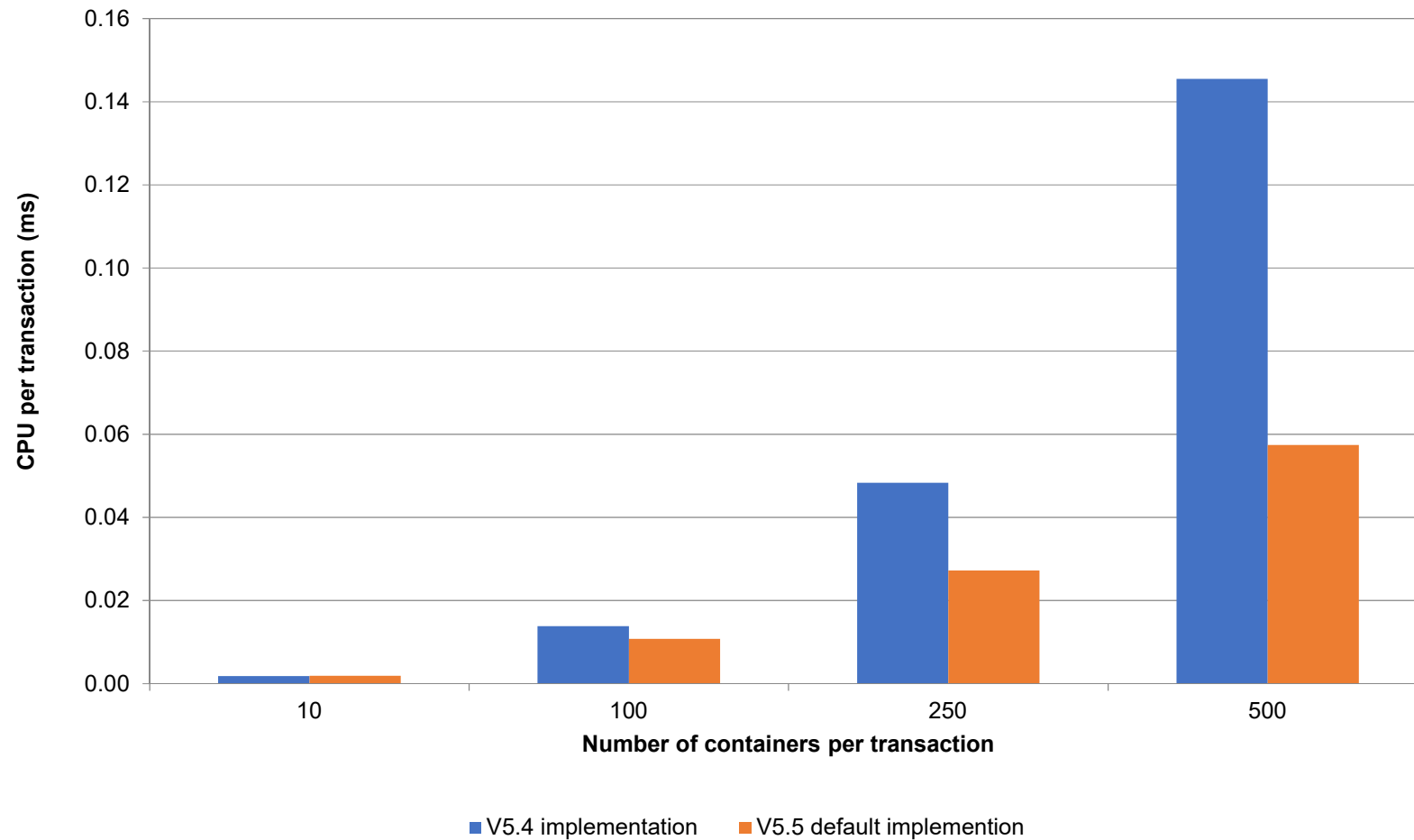
Notes

Improved performance for large number of containers on a channel

GET CONTAINER(NAME) searches a linked list for the container which impact performance for larger number of containers

v5.5 now use multiple shorter lists with a simple hash of the container name to select a list.

Channels performance improvements - CPU

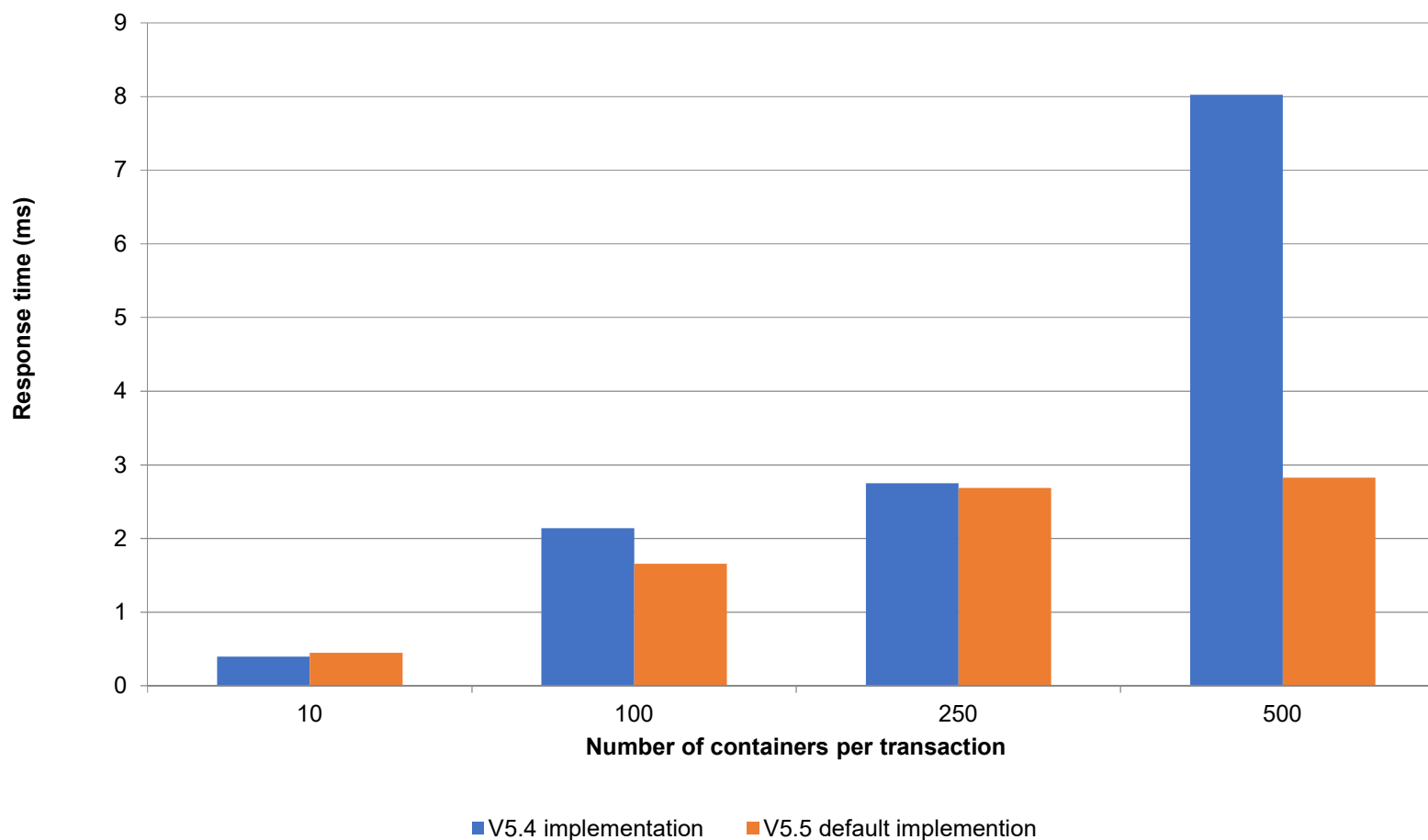


Notes

Containers are named blocks of data, designed for passing information between programs. Programs can pass any number of containers between each other. Containers are grouped in sets that are called *channels*. A channel is analogous to a parameter list. The CICS TS V5.5 release introduces a performance improvement which benefits applications where many containers are stored in a single channel.

CPU time for accessing larger number containers improved significantly.

Channels performance improvements – response time

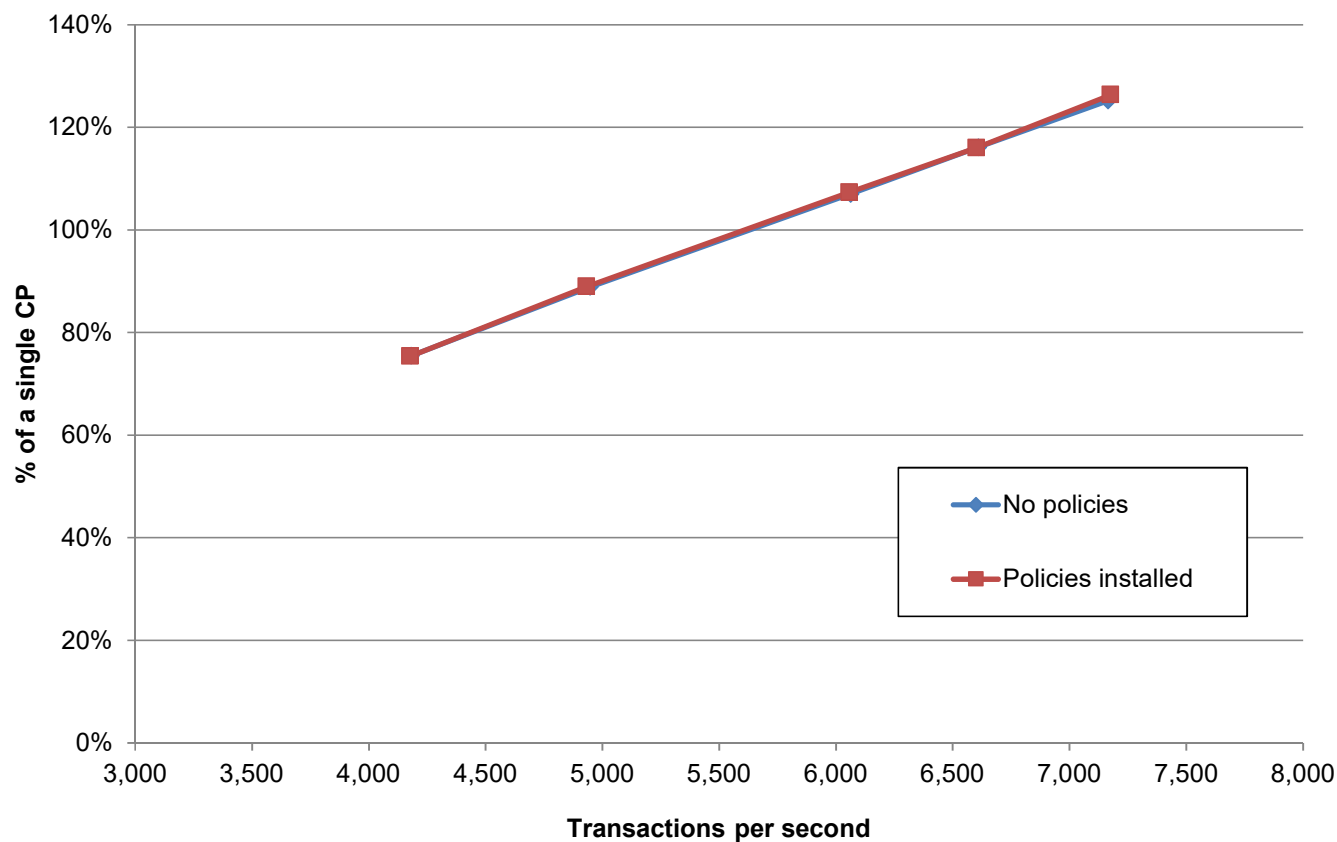


Notes

This chart shows the response time.

Policy performance impact

- To measure the overhead of enabling task rules
 - not to measure the overhead executing the action.
 - 19 task policies were installed but none is triggered.



Notes

The CPU per transaction is equivalent within measurable limits.
CICS continues to scale linearly as the transaction rate increases.

CICS manages release of USS process

- CICS TS V5.5 now manages the release of USS (UNIX System Services) processes from X8, X9, L8, and L9 TCBs.
 - If applications running on open TCBs use USS APIs, then a USS process is associated with the open TCB.
- The performance overhead of this additional USS process management
 - A simple socket application with minimum business logic was used.
 - This overhead was measured to be approximately **410 μ s** of CPU per task.
 - ~ half of the CPU overhead occurs in the CICS address space, and the remainder occurs in the OMVS address space.
 - Of the CPU overhead in CICS, ~ half of that is observed in the CICS performance class monitoring records.

Notes

Installation of PIPELINE and WEBSERVICE resources will result in USS process being dubbed to TCB.

Prior to this change, for L8, L9, X8 and X9 TCB the USS process remained with the TCB until the TCB terminated.

L8 mode TCBs are used by CONCURRENCY(THREADSAFE) and CONCURRENCY(REQUIRED).

L8 mode TCBs are used for CICSKEY OPENAPI application programs.

L9 mode TCBs are used for USERKEY OPENAPI application programs.

X8 and X9 run C and C++ programs compiled with the XPLINK option.

X8 TCBs are used for programs in CICS key.

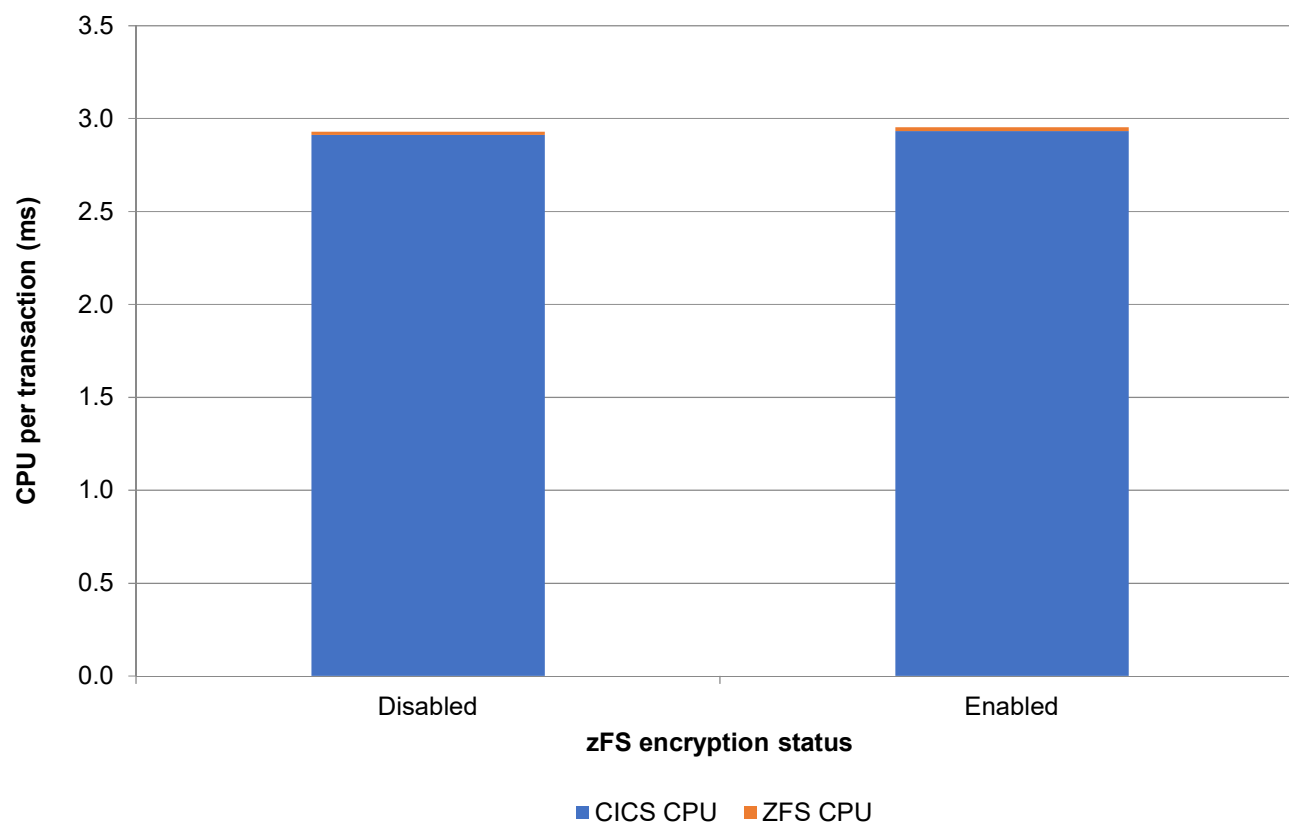
X9 mode TCBs are used for programs in user key.

zFS encryption performance

- In z/OS V2.3 zFS added support for encrypting file system data using DFSMS access method encryption.
- A WebSphere Liberty workload was used to test the performance overhead using an encrypted zFS file system.
 - 1,650 requests per second
 - ~30 MB of zFS data with SJ=ALL trace.

Notes

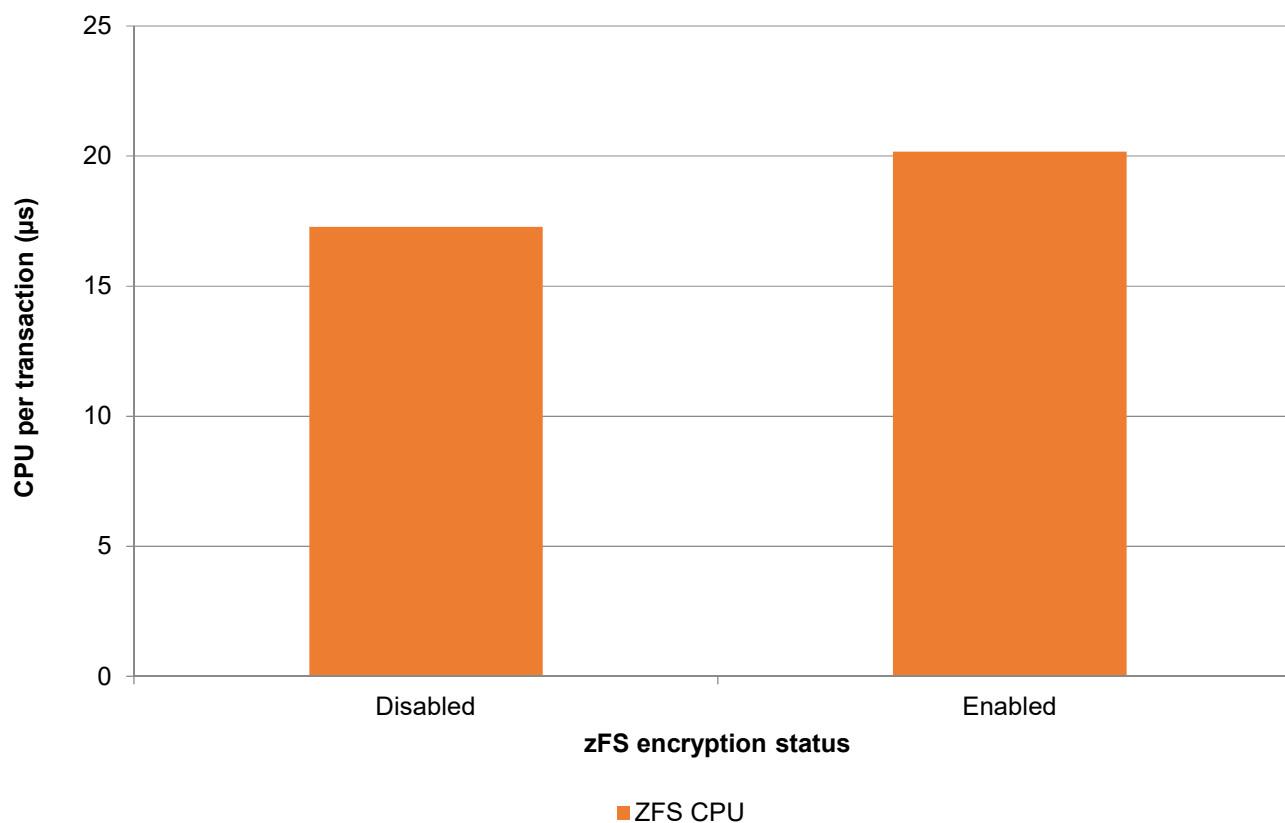
zFS encryption performance – CPU per transaction



Notes

The CPU consumed by the ZFS address space is a very small fraction of the overall CPU consumption per request

zFS encryption performance – zFS CPU per transaction



Notes

To more clearly demonstrate the difference in CPU attributed to the ZFS address space when enabling zFS encryption, only the ZFS address space data is plotted.

Although the zFS address space showed a significant relative increase in CPU cost per request (+17%), the overall total cost to the workload was negligible.

Improved instrumentation

Notes

Each release of CICS introduces additional areas of monitoring for performance analysis.

Resource class monitoring records (V5.5)

- New resource class monitoring records
 - Multiple resources per task
- URIMAP
 - Name, cipher, open / send / receive timings
- WEBSERVICE
 - Name, PIPELINE, INVOKE timings

Notes

Clients may now monitor, in real time, the URIMAPs and WEBSERVICES that are opened or invoked by CICS TS as a web client. CICS TS monitoring is enhanced with new monitoring records URIMAP and WEBSERVICE in the resource monitoring class. Multiple URIMAP or WEBSERVICE records can be monitored for one task.

A URIMAP record monitors the completion of WEB OPEN URIMAP, WEB RECEIVE, WEB SEND, and WEB CONVERSE requests that are issued by the user task for a URIMAP.

A WEBSERVICE record monitors the completion of INVOKE SERVICE requests that are issued by the user task for a WEBSERVICE, and tracks the name of the PIPELINE resource definition that was used.

This enhancement makes it easier to identify the URIMAPs or WEBSERVICES associated with prolonged socket wait time and diagnose troublesome destinations.

https://www.ibm.com/support/knowledgecenter/SSGMCP_5.5.0/reference/monitoring/dfht3_mon_tranmnr_fields.html

Monitoring and statistics (V5.5)

- Performance class
 - SSL cipher used for outbound requests
 - First message from a client (DFH SOCK)
 - Outbound web support in DFHWEBB and DFHWEBC groups
- Statistics
 - CICS policy rules
 - Transaction abend count
 - Peak aids in chain in ISC/IRC system entry

```

Policy name. . . . . : file_v51
Policy user tag. . . . . :
Bundle name. . . . . : PLCY51FC
Bundle directory . . . . . : /u/iburnet/git/cics-perf-workload-dsw-lsr/bu
                               : ndles/com.ibm.cics.perf.workload.dsw.lsr.pol
                               : icy.V51.file/
Rule name. . . . . : READ
Rule type. . . . . : filerequest
Rule subtype . . . . . : read
Action type. . . . . : abend
Action count . . . . . : 0
Action time. . . . . :
  
```

Notes

Outbound SSL cipher

The SOCIPHER field in the DFH SOCK group now reflects the SSL cipher used on outbound requests, in addition to inbound requests.

First message from a client

The SOCONMSG field in the DFH SOCK group indicates whether the task processed the first message for establishing a new connection for a client. This field helps you measure how often a new socket connection is created.

Outbound web support

Three new fields have been added in DFHWEBB to provide information on timing for each of the following commands:

- WBURIOPN – WEB OPEN URIMAP
- WBURIRCV – WEB RECEIVE and the receive portion of WEB CONVERSE
- WBURISND – WEB SEND and the send portion of WEB CONVERSE

CICS policy rules statistics

Statistics are now available for CICS policy rules. CICS collects resource statistics for each rule that is defined in a policy, and supplies a summary report.

Transaction abend count statistics

Statistics now displays the number of abends by transaction ID.

Performance publications

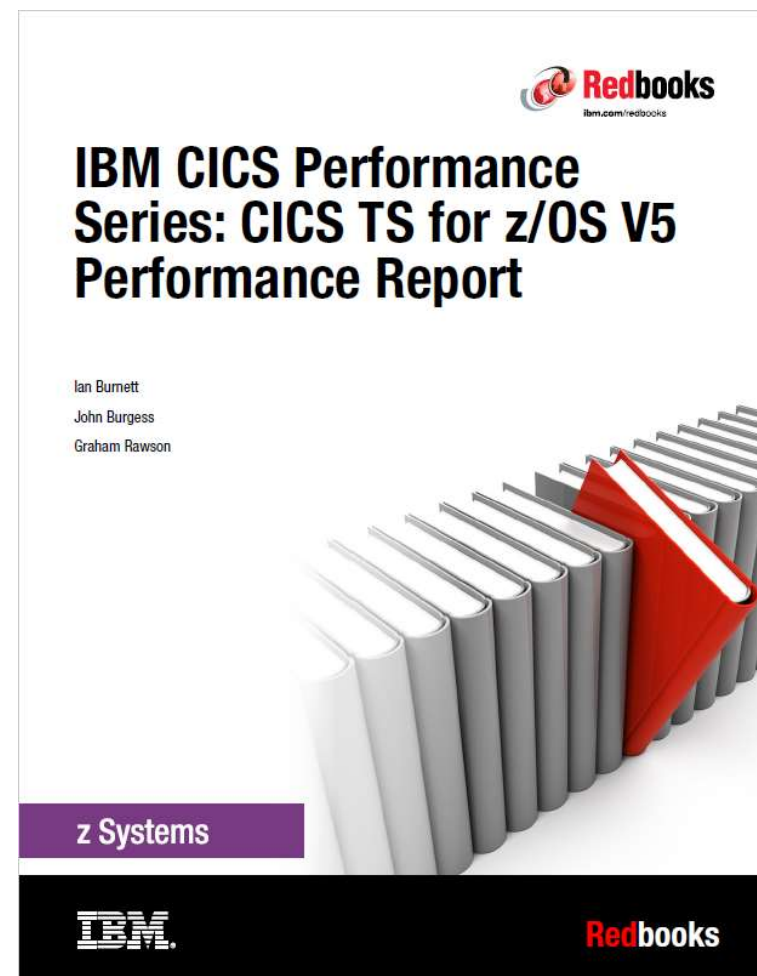
Notes

This page intentionally left blank.

CICS TS for z/OS Performance Report

- Part 1 – Performance concepts
 - Performance terminology
 - Test methodology and workload descriptions
 - Open transaction environment (OTE)
- Part 2 – Performance detail and measurements
 - CICS TS V5.1, V5.2, V5.3, V5.4, and V5.5 performance data
 - Comparisons to previous CICS releases
 - Monitoring, statistics, threadsafe enhancements
 - Performance-related SIT parameter changes

www.redbooks.ibm.com/abstracts/sg248298.html



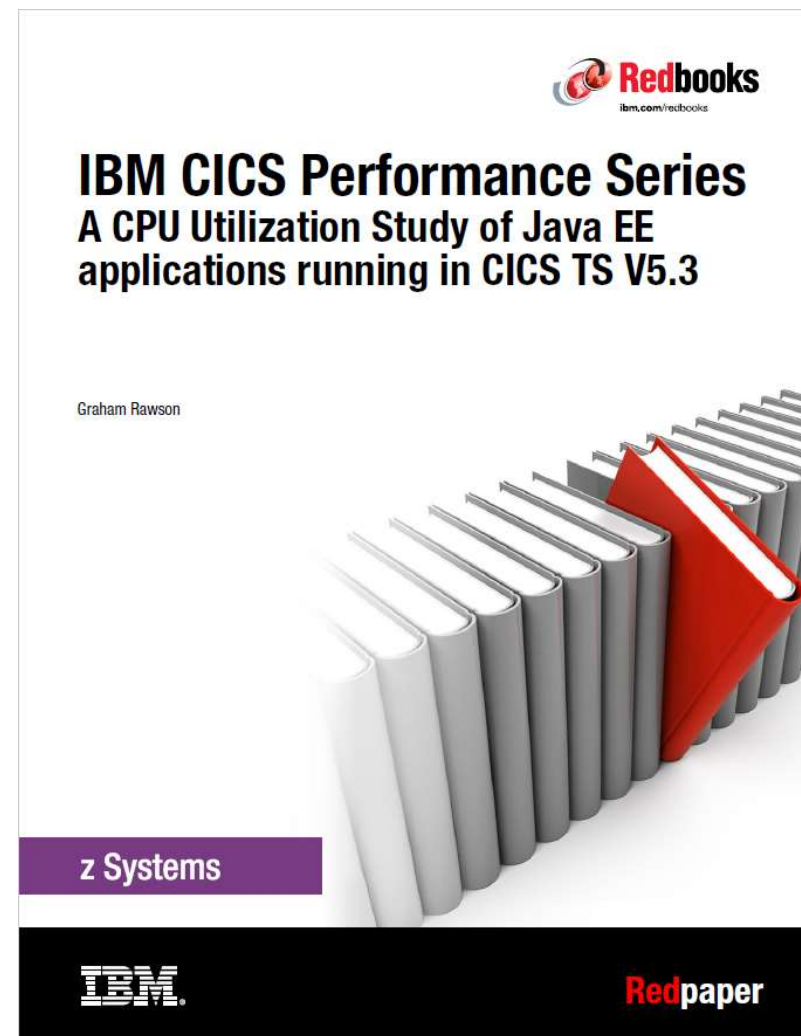
Notes

This page intentionally left blank.

IBM CICS Performance Series

- CICS TS for z/OS V5 Performance Report
- A CPU Utilization Study of Java EE applications
- CICS TS V5.3 Benchmark on IBM z13
- Web Services Performance in CICS V5.3
- CICS Type 2 and type 4 JDBC Driver Performance
- IBM CICS Interdependency Analyzer
- Plus others ...

ibm.biz/cicsredbooks



Notes

This page intentionally left blank.

Notices and disclaimers

- This presentation is provided by IBM Corporation. Copyright © IBM Corporation, 2018. Use and distribution by SHARE, Inc. permitted by license. Redistribution is prohibited.
- U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. This document is distributed “as is” without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.
- Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer’s responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer’s business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and disclaimers continued

- 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. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular, purpose.
- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.
- IBM, the IBM logo, ibm.com, AIX, BigInsights, Bluemix, CICS, Easy Tier, FlashCopy, FlashSystem, GDPS, GPFS, Guardium, HyperSwap, IBM Cloud Managed Services, IBM Elastic Storage, IBM FlashCore, IBM FlashSystem, IBM MobileFirst, IBM Power Systems, IBM PureSystems, IBM Spectrum, IBM Spectrum Accelerate, IBM Spectrum Archive, IBM Spectrum Control, IBM Spectrum Protect, IBM Spectrum Scale, IBM Spectrum Storage, IBM Spectrum Virtualize, IBM Watson, IBM Z, IBM z Systems, IBM z13, IBM z14, IMS, InfoSphere, Linear Tape File System, OMEGAMON, OpenPower, Parallel Sysplex, Power, POWER, POWER4, POWER7, POWER8, Power Series, Power Systems, Power Systems Software, PowerHA, PowerLinux, PowerVM, PureApplication, RACF, Real-time Compression, Redbooks, RMF, SPSS, Storwize, Symphony, SystemMirror, System Storage, Tivoli, WebSphere, XIV, z Systems, z/OS, z/VM, z/VSE, zEnterprise and zSecure are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Please submit your session feedback!

- Do it online at <http://conferences.gse.org.uk/2019/feedback/GE>

- This session is **GE**



1. What is your conference registration number?

💡 This is the three digit number on the bottom of your delegate badge

2. Was the length of this presentation correct?

💡 1 to 4 = "Too Short" 5 = "OK" 6-9 = "Too Long"

1 2 3 4 5 6 7 8 9

3. Did this presentation meet your requirements?

💡 1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

4. Was the session content what you expected?

💡 1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

Thank you



Jenny He

hejen@uk.ibm.com

CICS development, IBM Hursley Lab, UK

