GSE GUIDE SHARE EUROPE UK REGION

GSE UK Conference 2018
Better, stronger, faster; The Mainframe….. the Machine!

CICS performance update

Jenny He
CICS TS development, IBM Hursley Lab, Winchester, UK

November 2018
Session GG

Notes

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 Java, zIIP eligibility, new commands, and upgrading to z14 helping you save money and improve throughput.

- 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.
- This session will mostly about performance enhancements in CICS TS V5.4 and V5.5.

Measurement process for CICS performance

Notes

This page intentionally left blank.

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

CHPIDs - channel path identifiers

DASD - direct access storage device

Hardware instrumentation services (HIS) is a function that collects hardware event data for processors in SMF records type 113, as well as a z/OS UNIX System Services output files. You can only use HIS for IBM System z10 or later machines.

Environment (V5.4 testing)

- Hardware
  - z13 2964 model NE1
    - LPAR with up to 32 dedicated CPs
    - Separate LPAR with 8 dedicated CPs for network driver
  - DASD DS8870
  - Internal Coupling Facility with ICP links

- Software
  - z/OS 2.2
  - DB2 V11
  - CICS TS V5.3 refresh 24th February 2017
  - CICS TS V5.4

Notes

This is the environment we used to test the final build of the CICS TS V5.4 release. The z14 was not available at the time – more on the z14 later.

Release to release comparisons (V5.4)

Notes

Many workloads are executed to ensure the performance of CICS is not degraded when upgrading from one release to another.

DSW dynamic 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 routed from 4 TORs to 10 AORs via CPSM
- File access is VSAM/RLS

Notes

This page intentionally left blank.

DSW workload – comparing V5.3 to V5.4

ETR	CICS %	ms/tran
6139.73	253.15%	0.412
8974.81	365.76%	0.408
12099.66	483.50%	0.400
16444.72	661.87%	0.402
21386.62	871.39%	0.407

CICS TS V5.3
Average CPU / tran = 0.406ms

ETR	CICS %	ms/tran
6140.75	252.93%	0.412
8968.03	366.44%	0.409
12119.32	491.94%	0.406
16433.46	667.78%	0.406
21282.97	871.85%	0.410

CICS TS V5.4
Average CPU / tran = 0.408ms

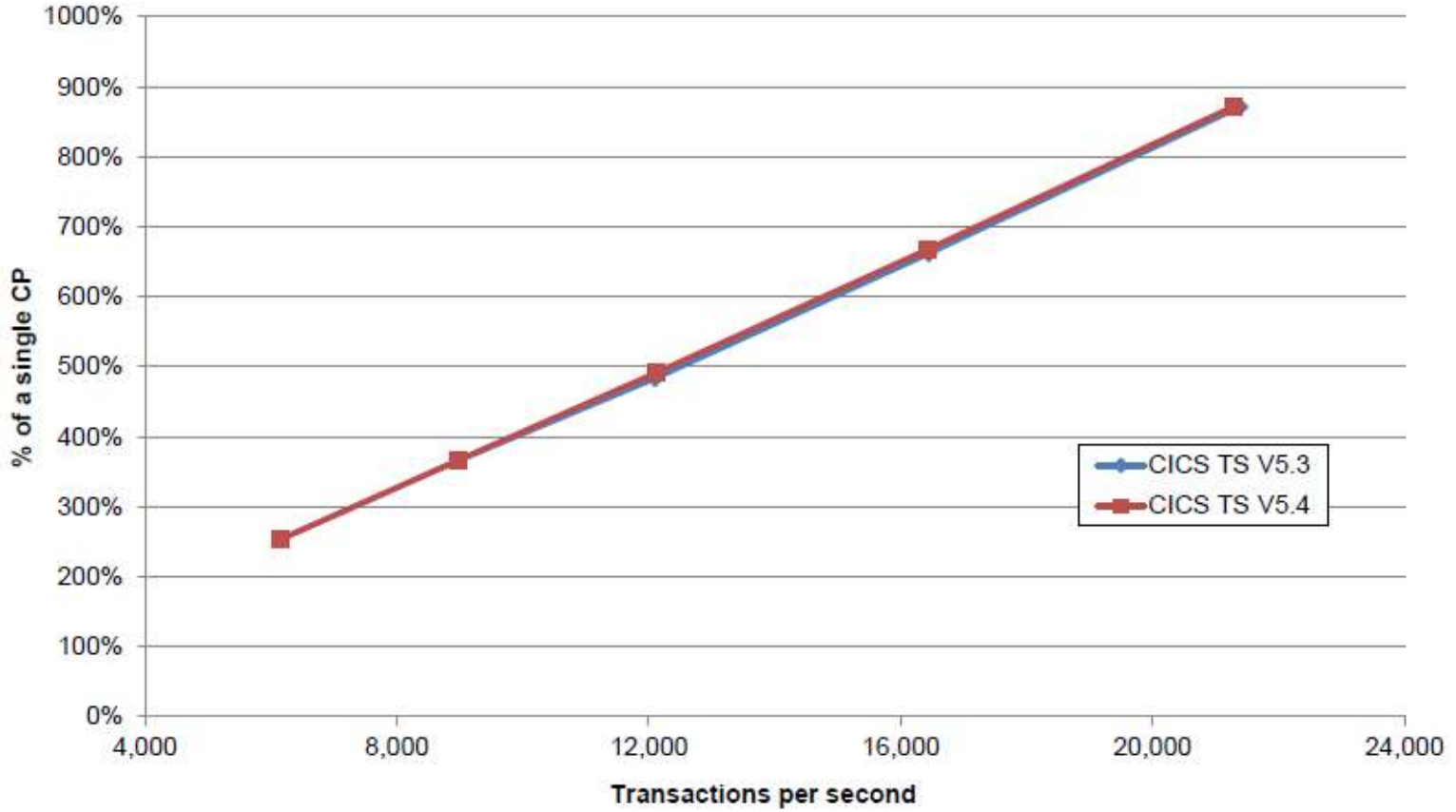
(< 0.5% delta)

Notes

external throughput rate (ETR) = the number of CICS tasks completed / elapsed time in second

CICS % : amount of CPU time that CICS region used.

DSW workload V5.3 to V5.4



Notes

This page intentionally left blank.

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

This page intentionally left blank.

RTW workload – comparing V5.3 to V5.4

ETR	CICS %	ms/tran
333.44	44.11%	1.323
499.59	66.11%	1.323
713.15	94.10%	1.319
996.18	131.52%	1.320
1241.70	163.99%	1.321

CICS TS V5.3
Average CPU / tran = 1.321ms

ETR	CICS %	ms/tran
333.47	44.47%	1.334
499.70	66.52%	1.331
713.23	94.15%	1.320
996.34	131.97%	1.325
1241.71	165.19%	1.330

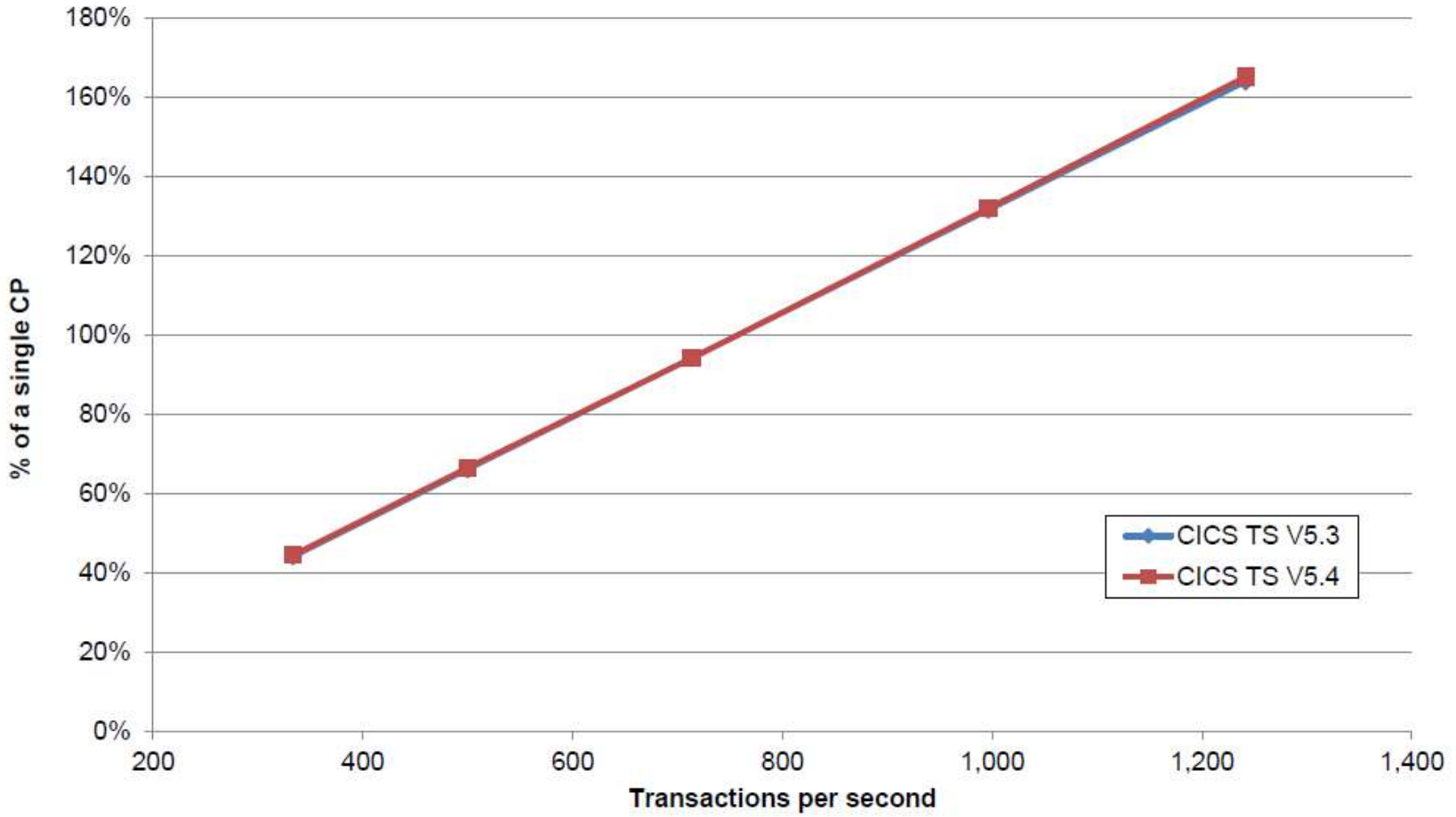
CICS TS V5.4
Average CPU / tran = 1.328ms

(< 0.5% delta)

Notes

The tables show RMF data extracted from 5 different transactions rates for both CICS TS V5.3 and CICS TS V5.4.

RTW workload V5.3 to V5.4



Notes

This page intentionally left blank.

Java enhancements

Notes

This page intentionally left blank.

Java hardware exploitation

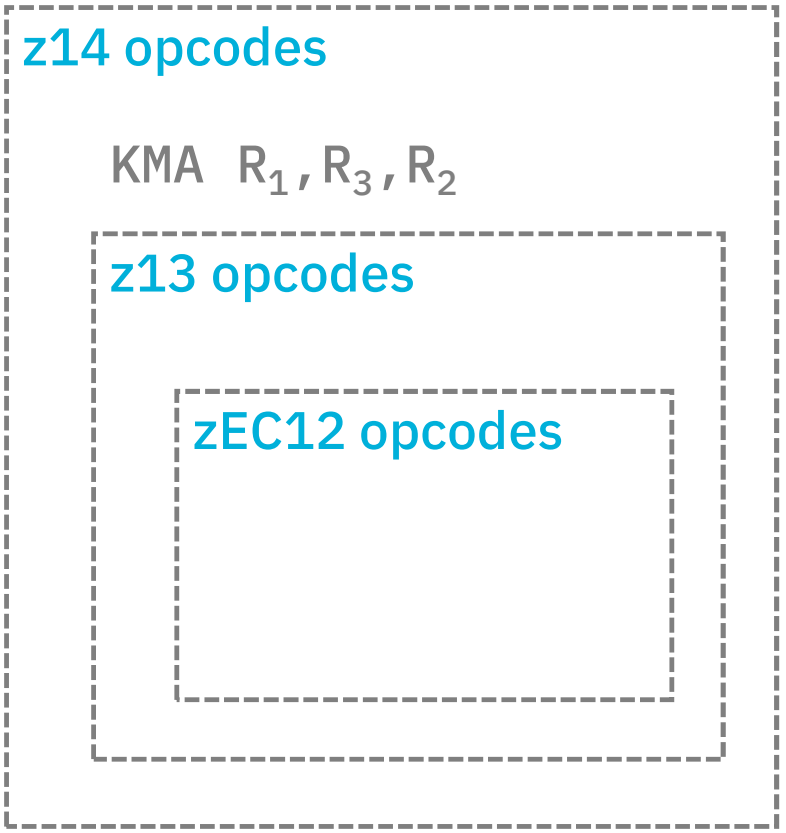
- Java 7.0 – z196 and zEC12 exploitation
  - New instructions
  - Transactional execution
- Java 7.1 – zEC12 exploitation
  - zEDC for zip acceleration
  - Increased zIIP offload from SR3 onwards
- Java 8.0 – language improvements and z13 exploitation
  - SIMD instructions
  - SMT mode 2
  - Use of crypto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

Notes

This page intentionally left blank.

IBM SDK for Java 7.0 vs 8.0

- IBM SDK for Java 7.0 SR10
  - Exploitation of zEC12 and earlier
- IBM SDK for Java 8 SR5
  - Exploitation of z14 and earlier



Notes

At runtime, the JVM compiles Java bytecode to native machine instructions. Later versions of the Java SDK for z/OS have knowledge of the instructions available in the latest hardware, can therefore make optimal use of the platform on which it is executing.

For example, the z14 adds extra function codes to the **COMPUTE INTERMEDIATE MESSAGE DIGEST** (**KIMD**) and **COMPUTE LAST MESSAGE DIGEST** (**KLMD**) hardware instructions, which are useful in modern cryptography algorithms. It also provides the new **CIPHER MESSAGE WITH AUTHENTICATION** (**KMA**) instruction for GCM-based encryption.

The most recent version of Z Systems hardware that IBM SDK for Java 7.0 SR10 can exploit is the IBM zEC12 and earlier.

IBM SDK for Java 8 SR5 exploits z14 hardware and earlier.

Java support by CICS

- Java 7
  - CICS V5.1, V5.2, V5.3, V5.4
- Java 7.1
  - CICS V5.1 + APAR PI30532
  - CICS V5.2, V5.3, V5.4
- Java 8
  - CICS V5.1 + APAR PI52819
  - CICS V5.2 + APAR PI52819
  - CICS V5.3 and later GA

Notes

This page intentionally left blank.

Java 8 recommended for CICS TS V5

CICS TS V5.5 requires IBM 64-bit SDK for z/OS, V8

IBM 64-bit SDK for z/OS, Java Technology Edition V7.0.0 and V7.1.0

Withdrawal Announcement 916-121 – August 2, 2016

To be withdrawn from service on September 30, 2019

WebSphere Application Server Liberty

WebSphere Application Server Liberty base – Removal Notices

Liberty fix pack 19.0.0.3: Liberty kernel will be recompiled and can no longer run with Java SE 7.

For details see Java 8 recommended for CICS TS V5

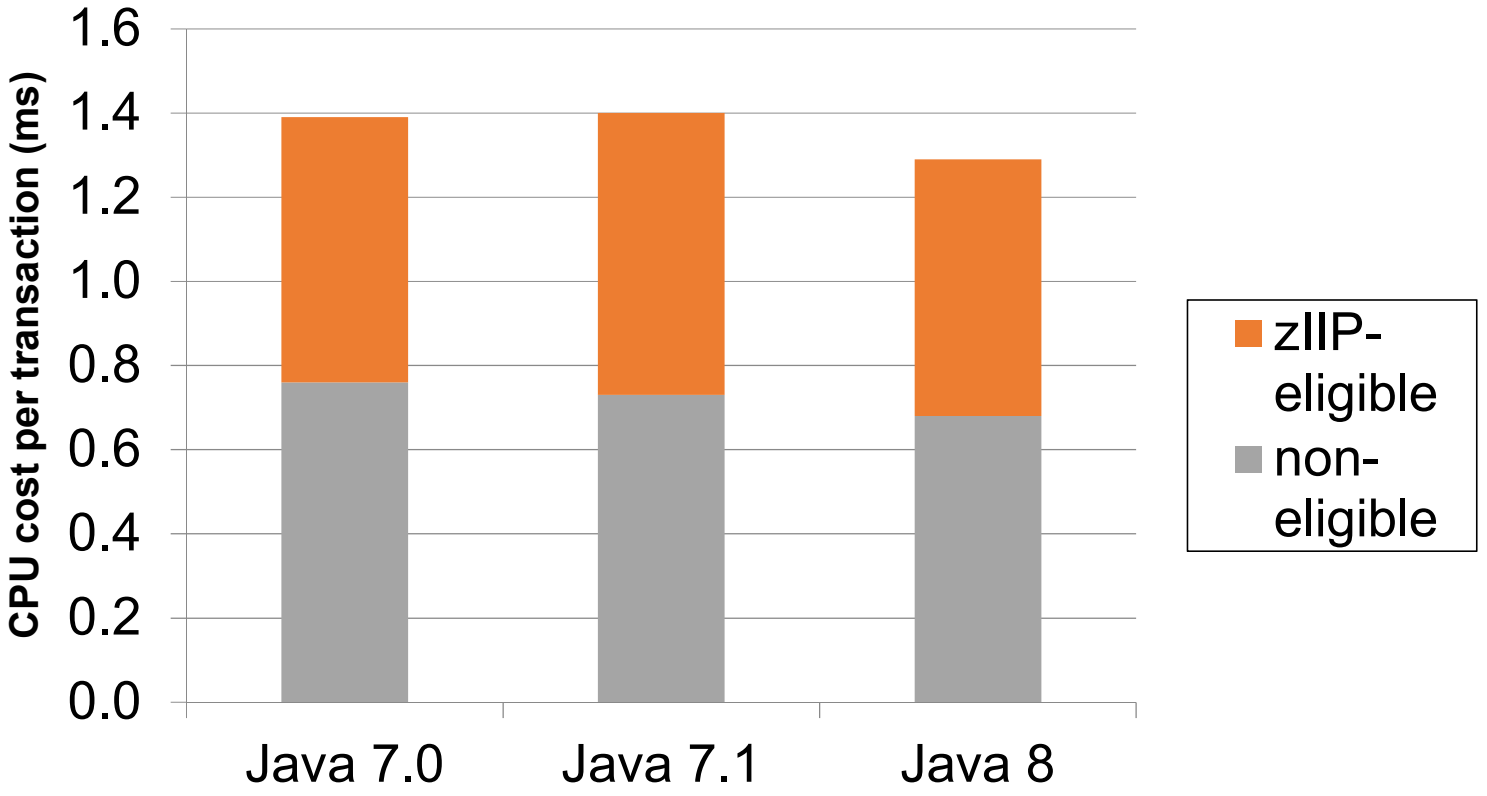
Notes

We have been recommending customers move to Java 8 for a while now.

CICS TS V5.5 and later will no longer support versions of the Java runtime earlier than Java 8.

https://developer.ibm.com/cics/2018/02/19/cics-support-ibm-sdk-java-technology-edition-version-8-service-refresh-5/

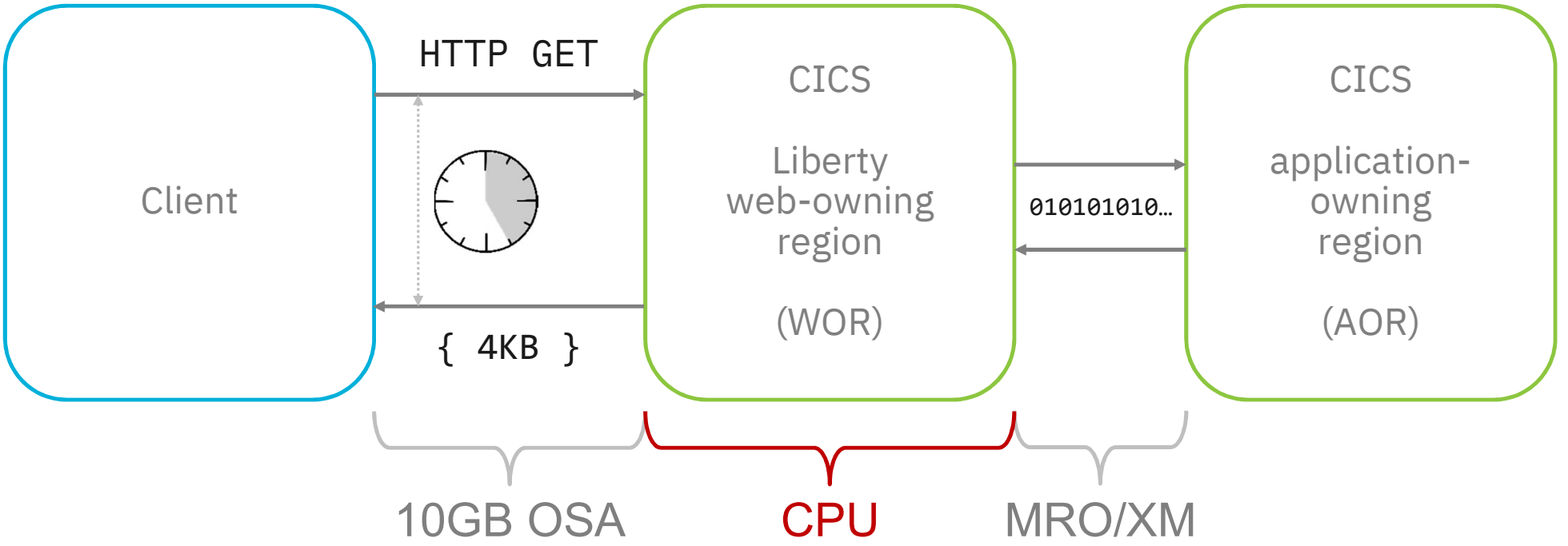
Performance of Java application in CICS



Notes

Chart uses an OSGi Java workload. Several applications provide a mixture of operations, including JDBC access, VSAM access, string manipulation, and mathematical operations.

SSL benchmark application configuration



Notes

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

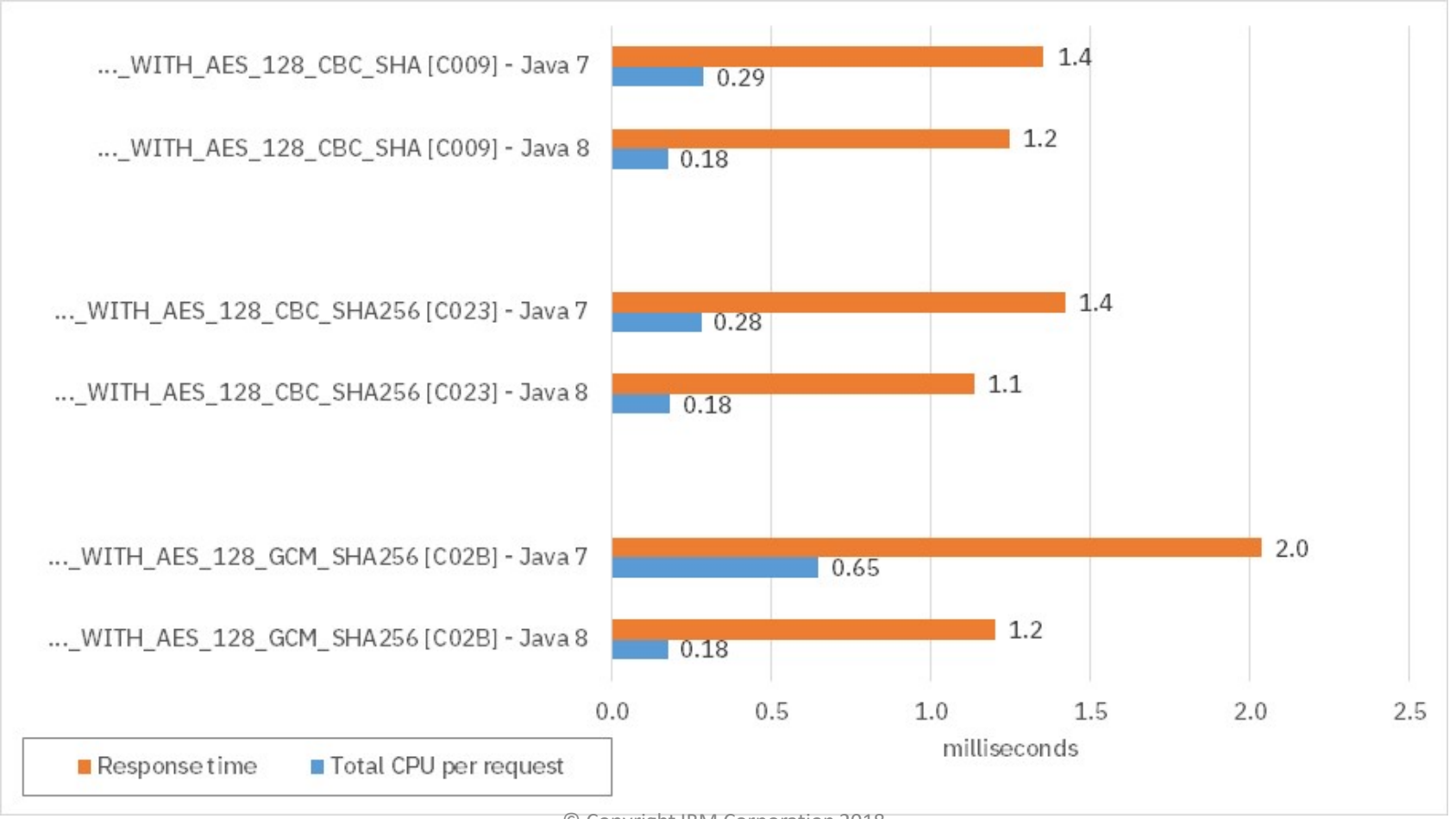
The WOR is running a simple JAX-RS application which then issues a link using the JCICS command to a program defined as remote in an application-owning region (AOR), connected using MRO/XM.

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.

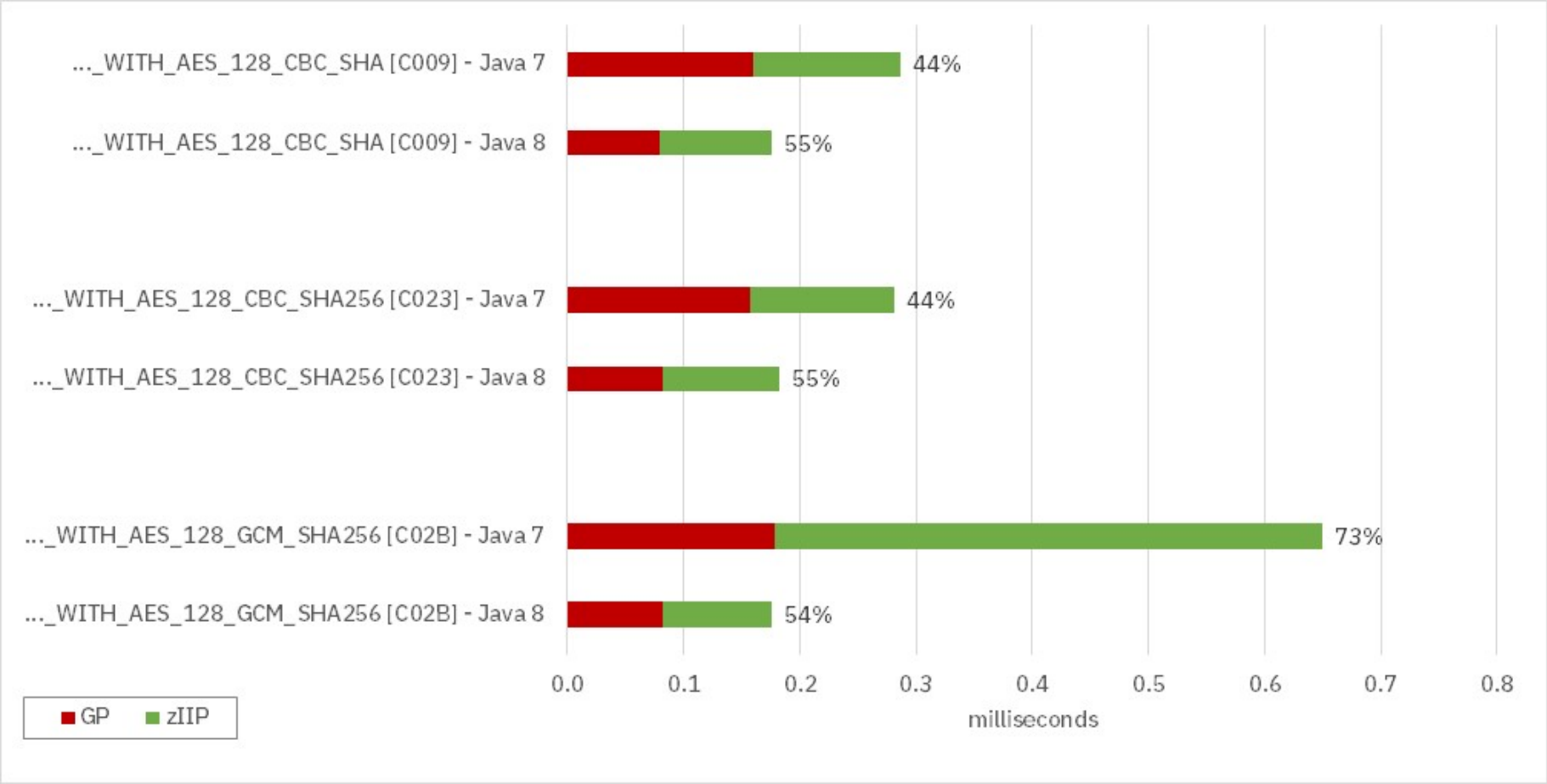
SSL and Java



Notes

This page intentionally left blank.

SSL and zIIP eligibility



Notes

This page intentionally left blank.

Liberty in CICS threading model

Notes

This page intentionally left blank.

Change to threading model for Liberty (V5.4)

- CICS TS V5.4
  - Also CICS TS V5.3 + APAR PI54263
- Utilises Liberty thread-pooling mechanism
  - A pool of T8 TCBs with just enough CICS context to be a CICS task.
  - Replace the old way of creating new CICS task every time.
- Transaction attach/detach now happens on T8 TCB
- Improvement in zIIP-eligibility rates

developer.ibm.com/cics/2016/05/17/improving-ziip-eligibilty/

Notes

The prior implementation for Liberty server embedded in CICS used the same scheme as our original OSGi JVM server where CICS would ‘lend’ the JVM a task to use as a thread each time a task wanted to run some Java. This meant there was no thread-pooling and every time Liberty wanted a thread, we’d attach a transaction, put it on a TCB from the CICS Dispatcher’s pool of T8s and then give it to the JVM.

This works fine, but thread-pooling is a good thing and Liberty has very nice schemes to optimize thread usage for its workloads.

In the new scheme, a T8 TCB with just enough CICS context to be a CICS task is given over to the Liberty threadpool.

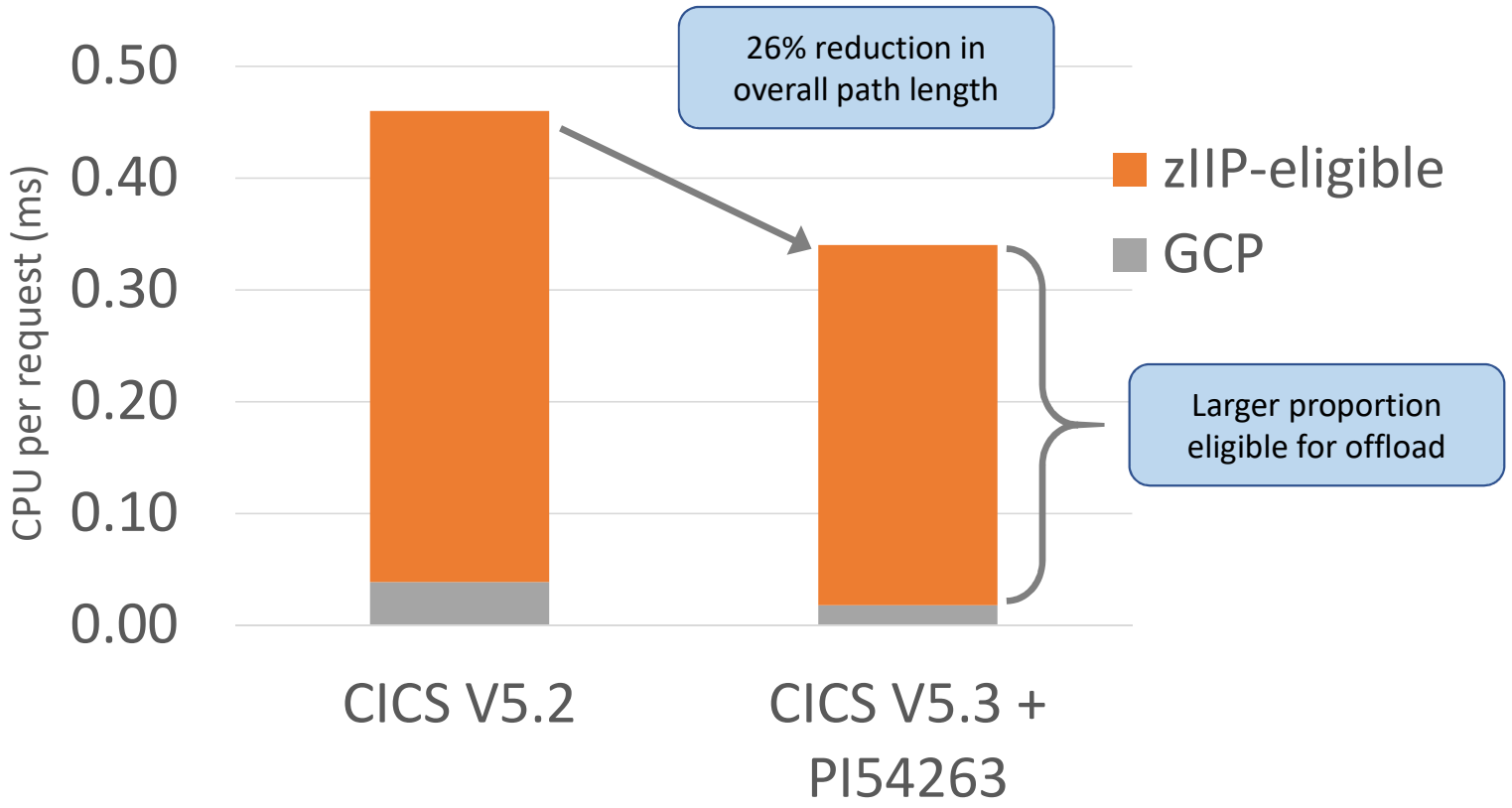
This task has no transaction context, no tranid, no task number, no security context, no unit-of-work…

but it can run Java code, enough for Liberty to manage it as a thread in its pool.

Also this task on a T8 would be dispatched on a zIIP processor since it lives inside the JVM.

When Liberty has some work for the thread, CICS will build the rest of the normal transaction environment before executing the application method.

zIIP-eligibility for Liberty



Notes

This page intentionally left blank.

Threadsafe improvements

Notes

This page intentionally left blank.

TCB Switching

CONCURRENCY	API		Initial TCB	DB2 or MQ command	Threadsafe command	Non-threadsafe command
QUASIRENT	CICSAPI		QR	QR → L8 → QR	no switch	no switch
THREADSAFE			QR	L8	no switch	QR
REQUIRED			L8	no switch	no switch	L8 → QR → L8
THREADSAFE	OPENAPI		L8	no switch	no switch	L8 → QR → L8
REQUIRED		stgprot (yes) execkey (user)	L9	L9 → L8 → L9	no switch	L9 → QR → L9

For Java programs a T8 TCB is used.
For an C xplink program a X8 TCB is used.

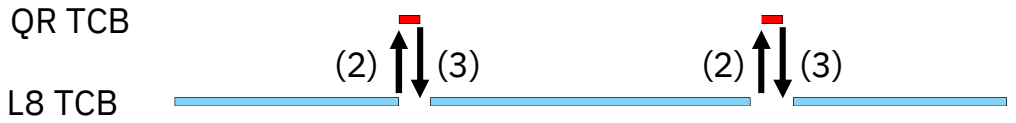
Notes

TCB switching in relation to program definition.

TCB switching due to non-threadsafe command

V4.1 – CONCURRENCY(THREADSAFE) with non-threadsafe TD

V4.2 – CONCURRENCY(REQUIRED) with non-threadsafe TD



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 demonstrate the initial TCB is QR then switching to L8 for DB2 command.

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

Threadsafe – impact on performance

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

V5.1

QR = 0.03ms

L8 = 6.17ms

8 TCB switches

Notes

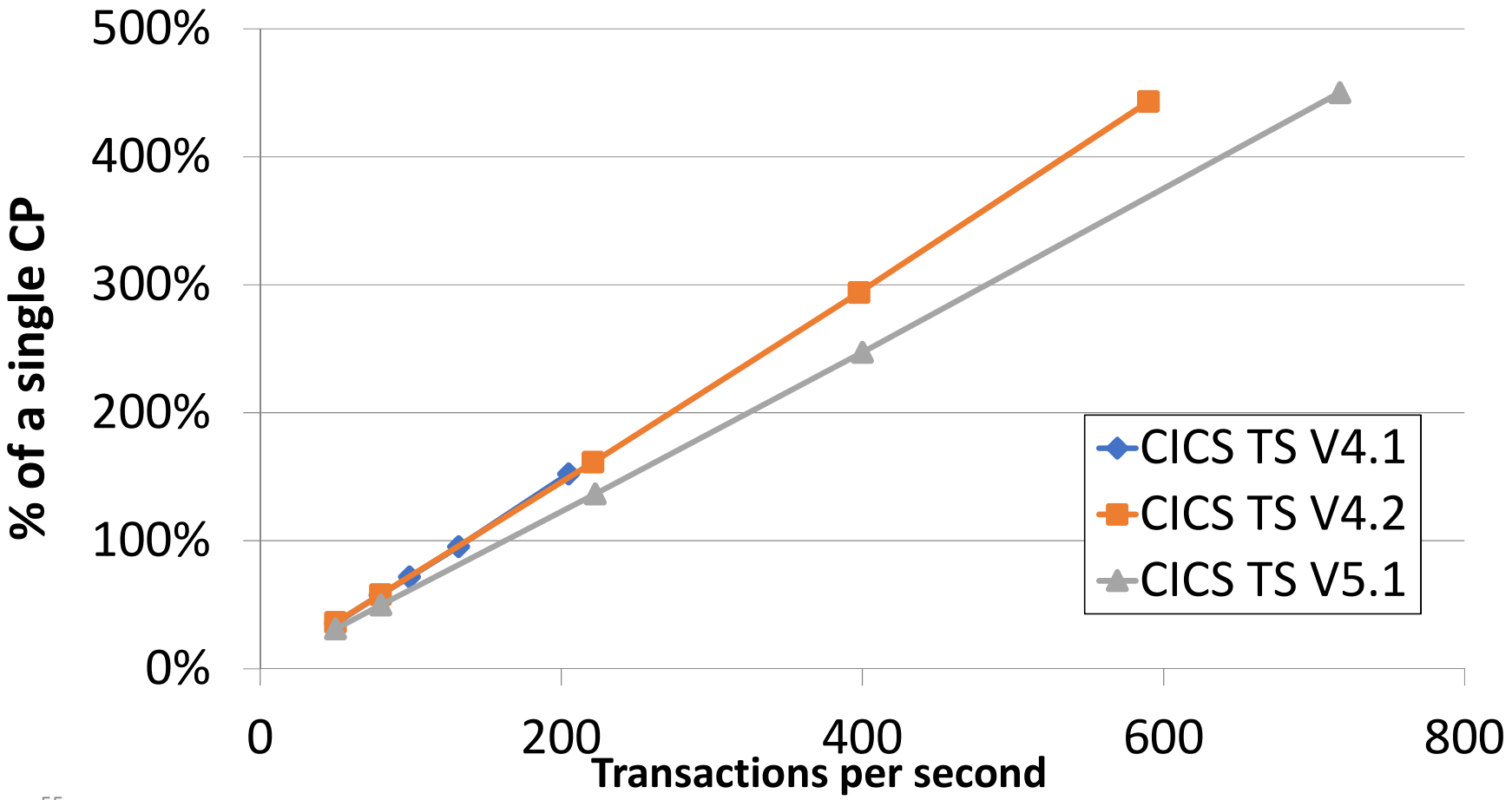
Chart shows a extracts from CICS Performance Analyzer reports for each of the various CICS TS levels.

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 t 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 ms / 4.60 ms/tran = 217 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 API (V5.4)

- New API commands in V5.4 are threadsafe:
  - `RUN TRANSID`
  - `FETCH ANY`
  - `FETCH CHILD`
  - `FREE CHILD`

  - `TRANSFORM DATATOJSON`
  - `TRANSFORM JSONTODATA`

Notes

This page intentionally left blank.

Threadsafe improvements (V5.5)

- Access to CFDTs is now threadsafe
  - READ, REWRITE, DELETE, WRITE, STARTBR, READNEXT, READPREV, ENDBR, RESETBR, UNLOCK commands for CFDTs can run on an open TCB as well as QR TCB.
  - Syncpoint also can occur on Open TCB
  - Initial open and load still occur on QR TCB
- EXEC CICS QUERY SECURITY
  - Reduce number of TCB switches to RO

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.

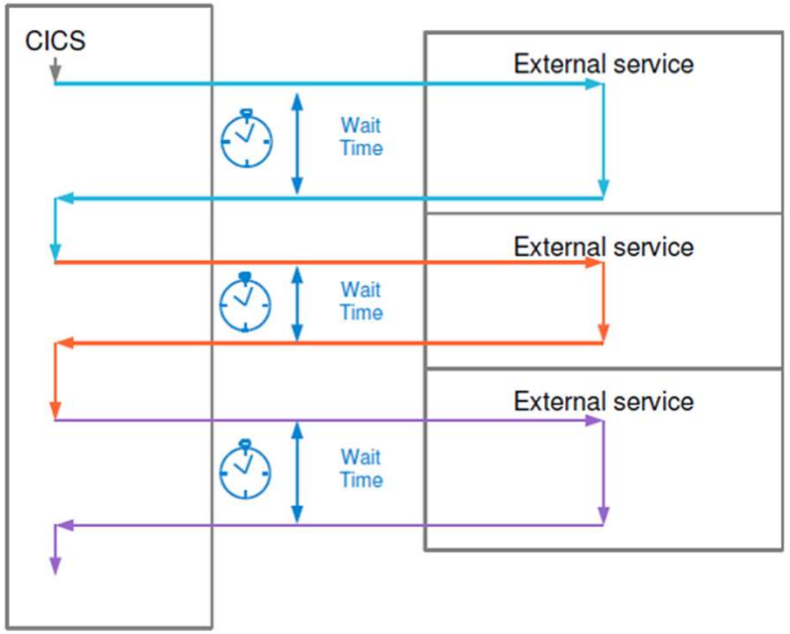
In earlier releases, CICS required multiple TCB switches to complete the EXEC CICS QUERY SECURITY call, and these have been optimized to just a pair of switches to the RO TCB and back.

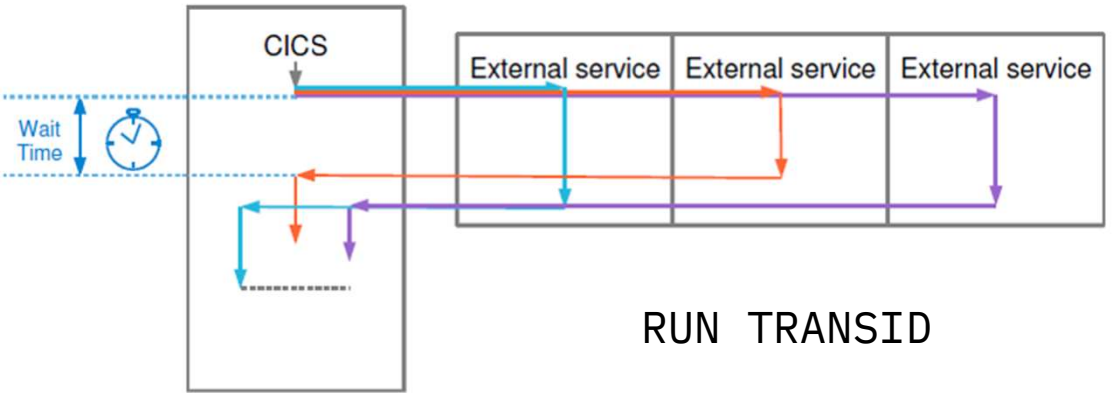
Asynchronous APIs (V5.4)

Notes

The Asynchronous API was introduced in CICS TS V5.4.

Wait for external services asynchronously






```
RUN TRANSID

FETCH CHILD

FETCH ANY

FREE CHILD
```

Notes

Could implement this yourself using IBM MQ, ECBs, EXEC CICS START, polling of common resource, …

Asynchronous API alternatives

- Application usage of ECBs or EXEC CICS ENQ & DEQ
  - Synchronize access to common storage
  - Task management code is application's responsibility
  - Timeouts difficult to code
  - Common storage and child tasks can become orphaned after abends
- Polling of CICS service (e.g. temporary storage)
  - Wasteful of CPU if polling time too short
  - Unnecessary delays if polling time too long

Notes

This page intentionally left blank.

Asynchronous API alternatives continued

- Notify on task completion using TD trigger queues
  - Less wasteful than polling
  - Still requires application management of tasks
  - Timeouts still problematic
- External resource manager such as IBM MQ
  - Calling external address space adds CPU overhead
  - Increases application complexity: increased management and monitoring overhead

Notes

This page intentionally left blank.

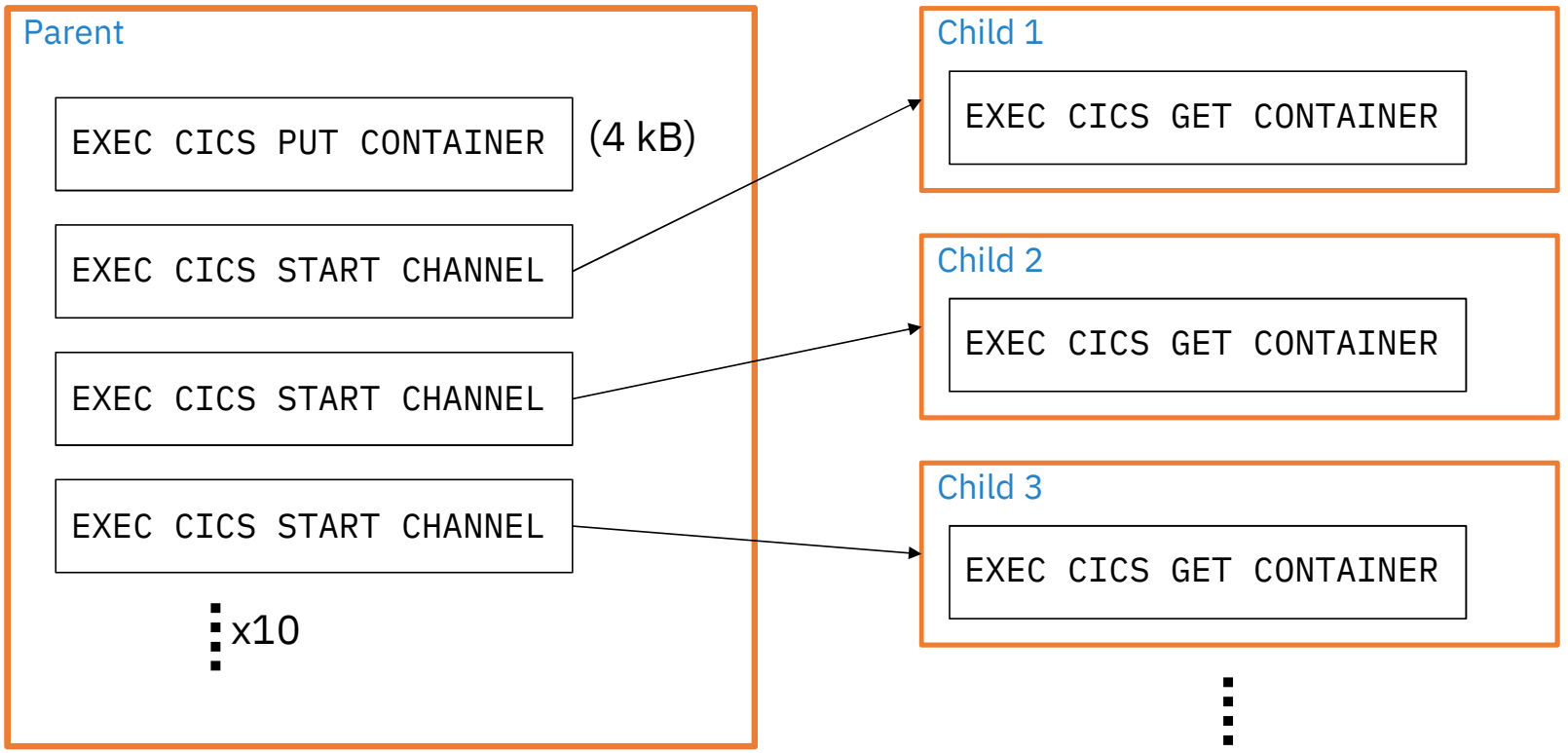
Asynchronous API (V5.4) performance comparison

- All similar mechanisms require a new task
- Compare `EXEC CICS START` with `EXEC CICS RUN TRANSID`

Notes

This page intentionally left blank.

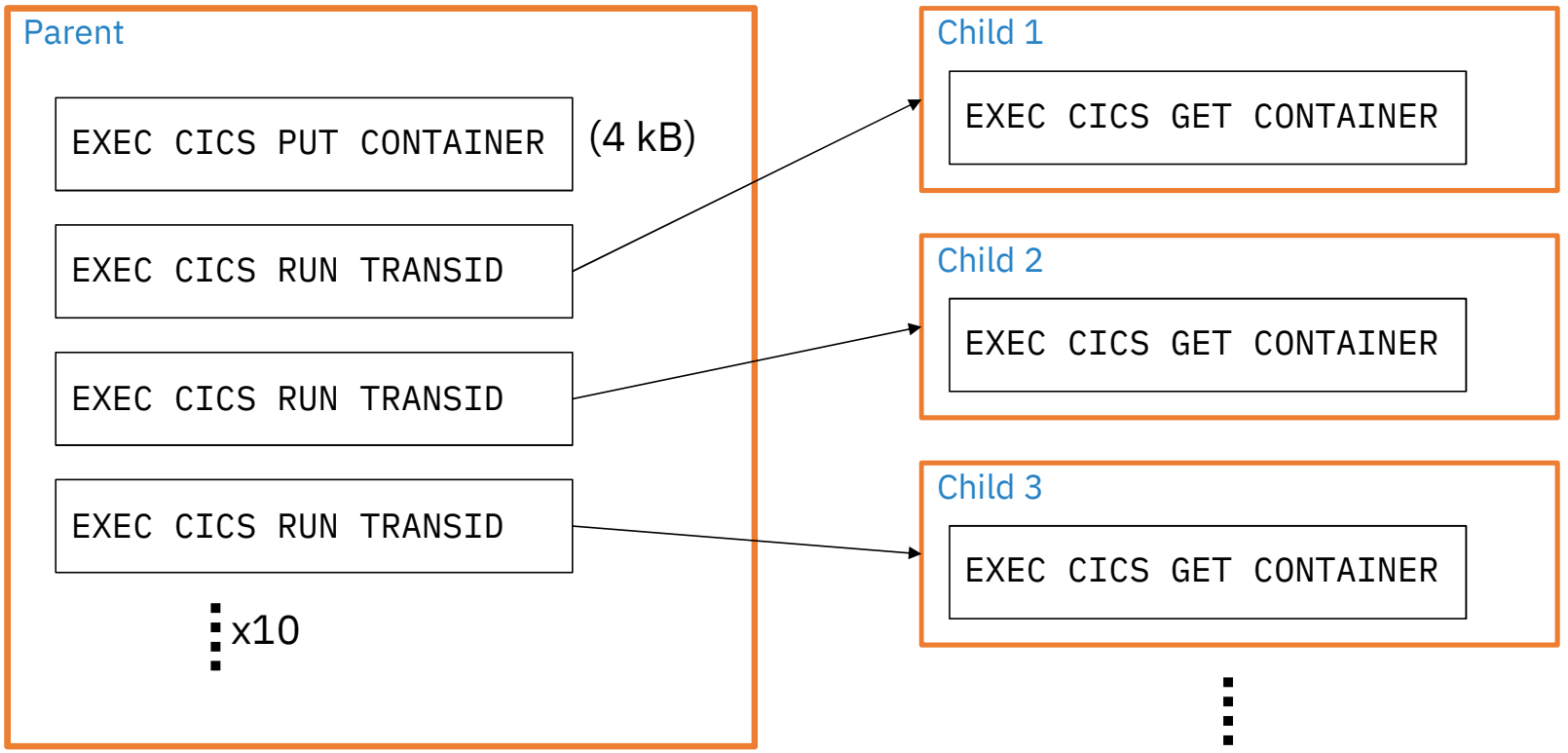
Application flow (EXEC CICS START)



Notes

This page intentionally left blank.

Application flow (EXEC CICS RUN TRANSID)



Notes

This page intentionally left blank.

Program definitions

- Parent programs defined CONCURRENCY(REQUIRED)
  - Force parent task to execute on L8 TCB
    - Parent task will switch to QR TCB when issuing START.
    - Parent task will stay on L8 TCB when issuing RUN TRANSID.
- Child programs defined CONCURRENCY(THREADSAFE)
  - Child program will start on QR TCB

Notes

This page intentionally left blank.

Performance results

	Avg. response time (ms)	Avg. CPU time (ms)	Avg. QR CPU time (ms)	Avg. QR dispatch time (ms)	TCB change mode operations
START	0.631	0.184	0.125	0.137	30
RUN TRANSID	0.284	0.140	0.030	0.038	10

- CPU and TCB change mode differs because:
  - START is non-threadsafe
  - RUN TRANSID is threadsafe

Notes

TCB change modes baseline:

- Initial program invocation (1)
- EXEC CICS RECEIVE (2)
- EXEC CICS SEND (2)
- Syncpoint and TRUE processing (4)
- Program terminate (1)

Asynchronous API benefits

- No more expensive than the cheapest method of starting a new task
- Fully-supported
- Workflow management
- Parent-child communication managed using 64-bit storage
- Uses existing and enhanced CICS statistics and monitoring data

More on Asynchronous API - Session GK – Tomorrow 10:45

Notes

This page intentionally left blank.

EXCI channels and containers

Notes

This page intentionally left blank.

External CICS Interface (EXCI) overview

- Call into CICS from batch application
- Call interface
  1. Allocate_Pipe
  2. Open_Pipe
  3. DPL_Request
  4. Close_Pipe
  5. Deallocate_Pipe
- EXEC CICS interface
  1. EXEC CICS LINK
- Limit of 32,763 bytes per DPL_Request or LINK call

Notes

This page intentionally left blank.

EXCI enhancements (V5.4)

- New API commands in EXCI environment
  - GET CONTAINER
  - PUT CONTAINER
  - MOVE CONTAINER
  - DELETE CONTAINER
  - DELETE CHANNEL
  - EXEC CICS LINK CHANNEL
  - DPL_Request – new version 3 of parameter list
- No (practical) limit on data transmitted in a single call
- Only EXCI libraries on V5.4 level – server on V3.1 or later

Notes

This page intentionally left blank.

New EXCI commands (V5.5)

- More new EXCI commands:
  - QUERY CHANNEL
  - STARTBROWSE CONTAINER
  - GETNEXT CONTAINER
  - ENDBROWSE CONTAINER

Notes

The external CICS interface (EXCI) provides four new commands:

- QUERY CHANNEL
- STARTBROWSE CONTAINER
- GETNEXT CONTAINER
- ENDBROWSE CONTAINER

These new commands enable EXCI users to query the number of containers on a channel and to browse the names of the containers on a channel.

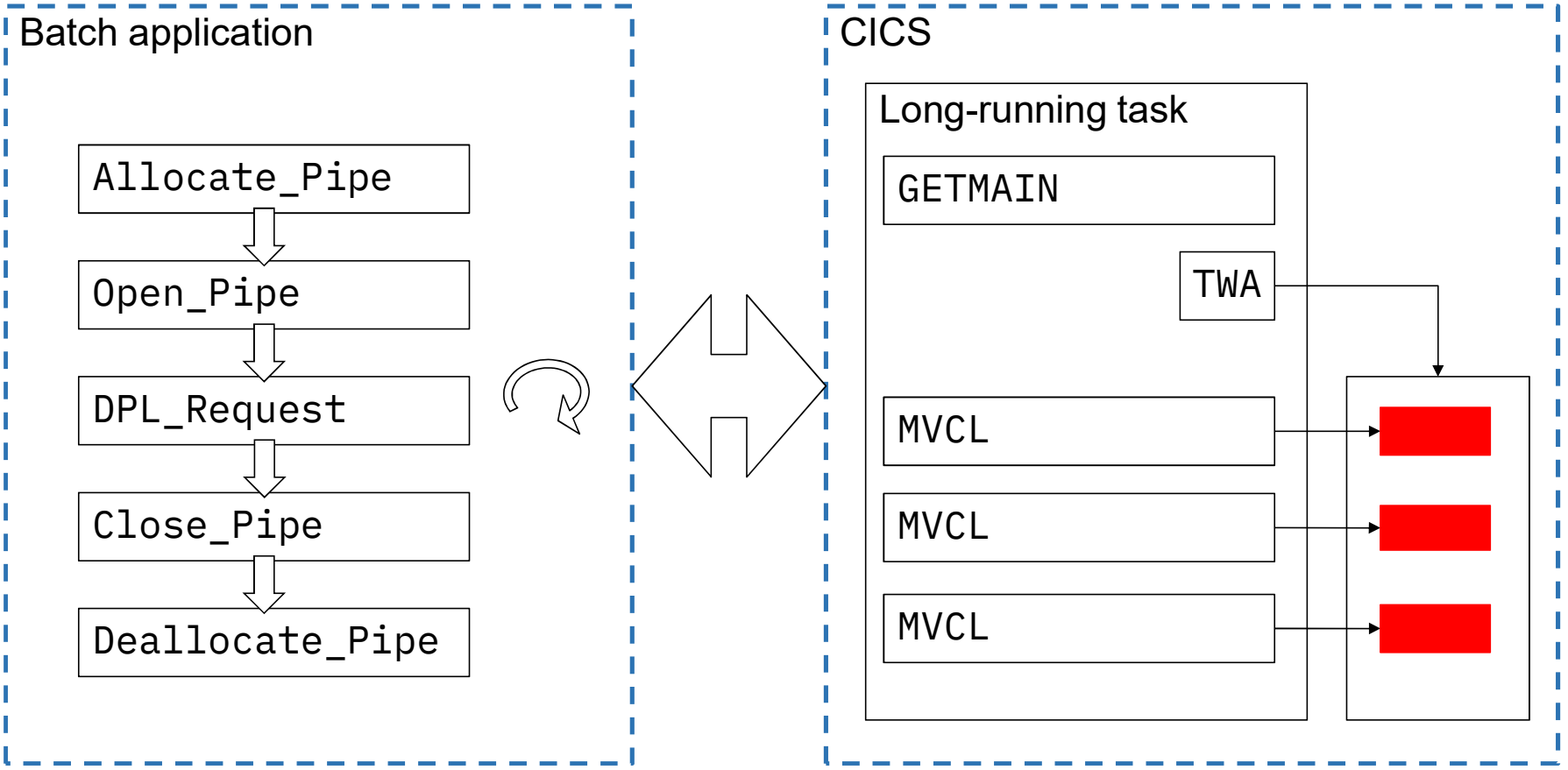
Performance comparison

- Sending data to CICS TS:
  1. Using ‘chunked’ data in COMMAREA
  2. Using single data area in a channel

Notes

This page intentionally left blank.

The application (COMMAREA)



Notes

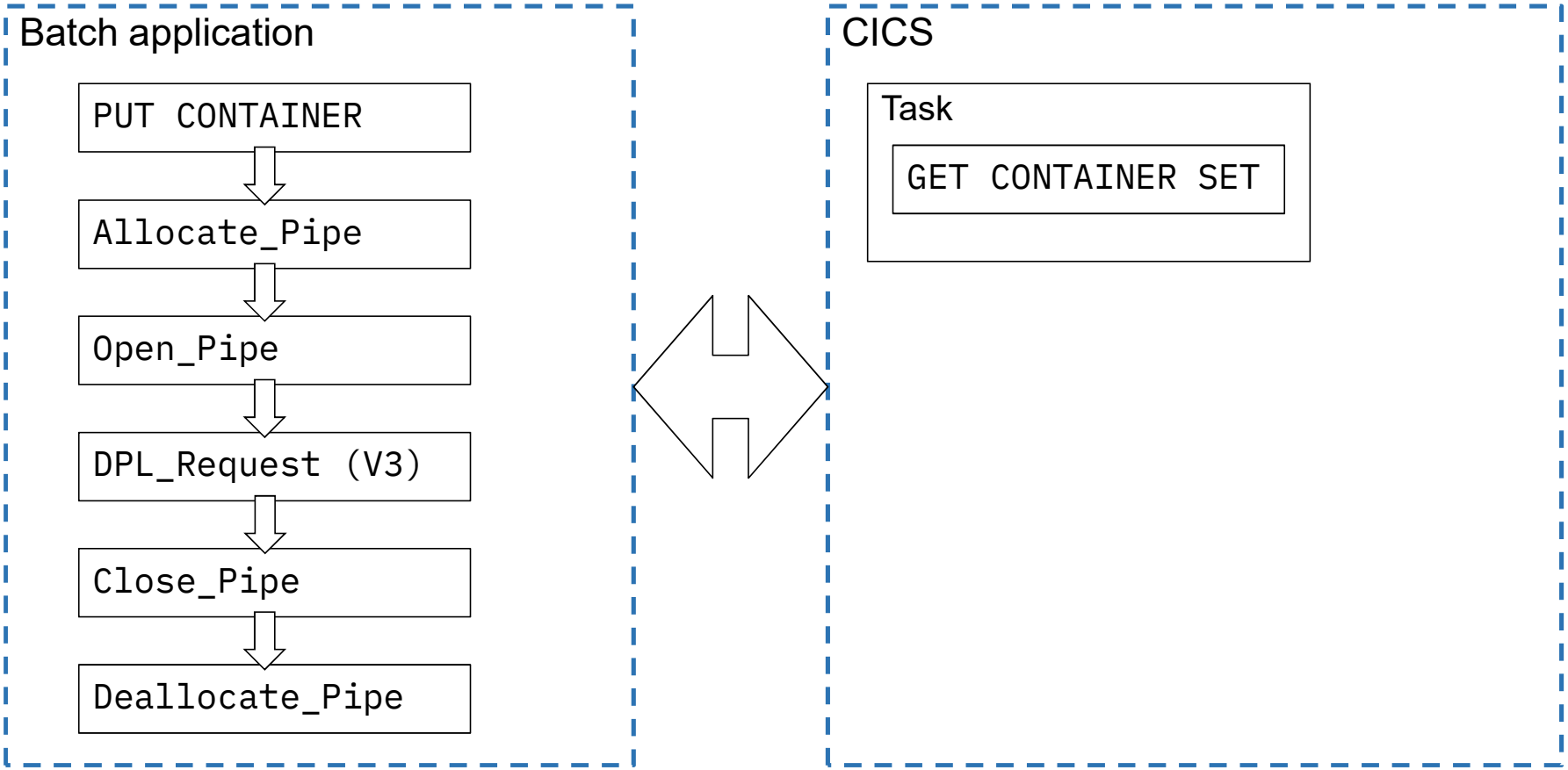
The client calls into the server using a DPL request.

The server has a long-running mirror task, and on first call allocates a storage area large enough for the whole payload. The location of this storage area is saved in the transaction work area (TWA).

Each request from the client is limited to 32 KB of data (maximum commarea size). Each call in will copy the data from the received commarea into the previously-allocated large storage area.

On the final flow, all the received data has been accumulated in the preallocated storage area, and the business logic can now commence.

The application (channels)

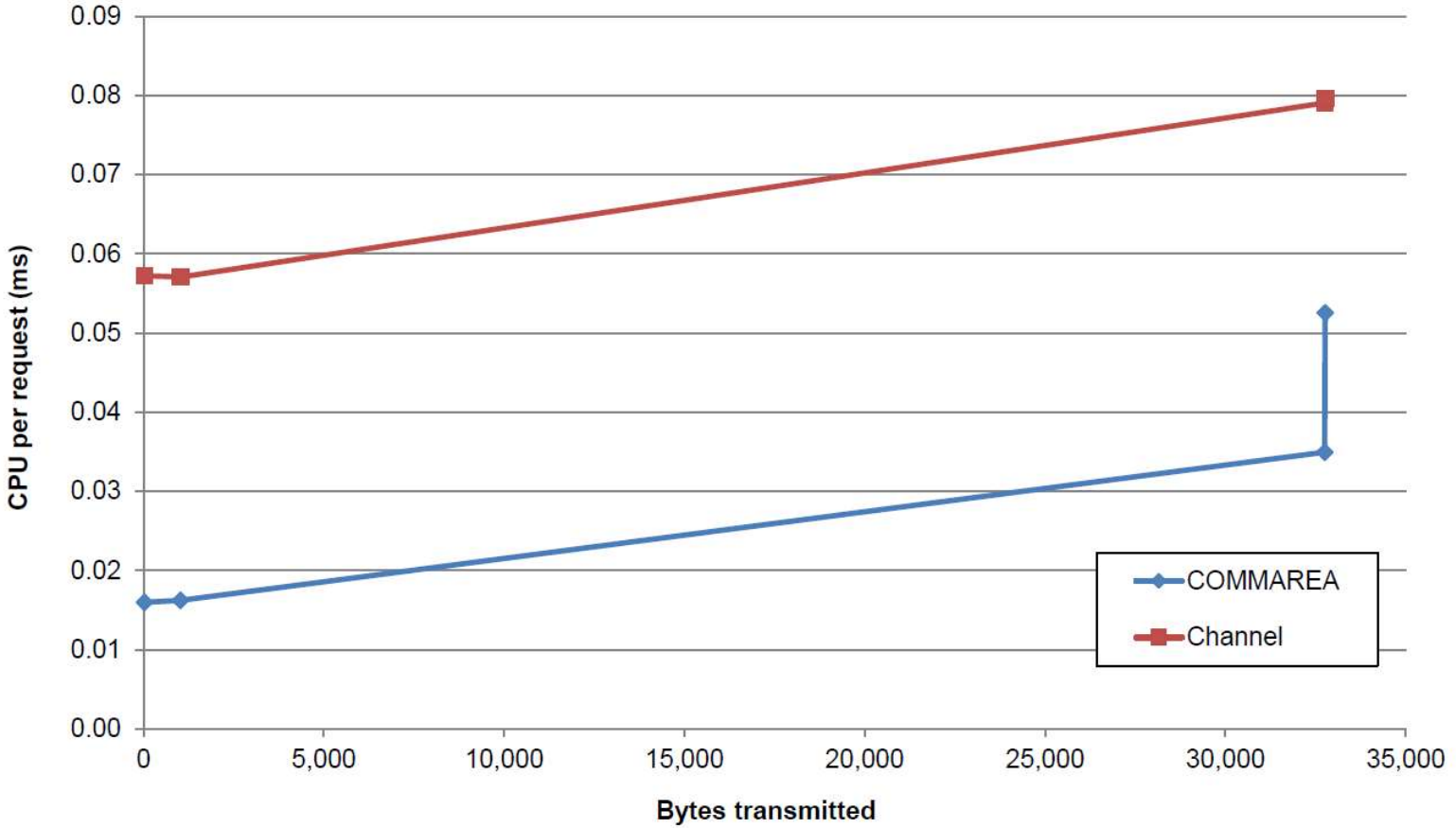


Notes

The use of channels and containers provides a much cleaner client and server side application.

There is no need for chunking of data, and the server simply receives the data area using the EXEC CICS GET CONTAINER SET call, which provides the required data in a CICS-managed storage area.

Small payload performance results

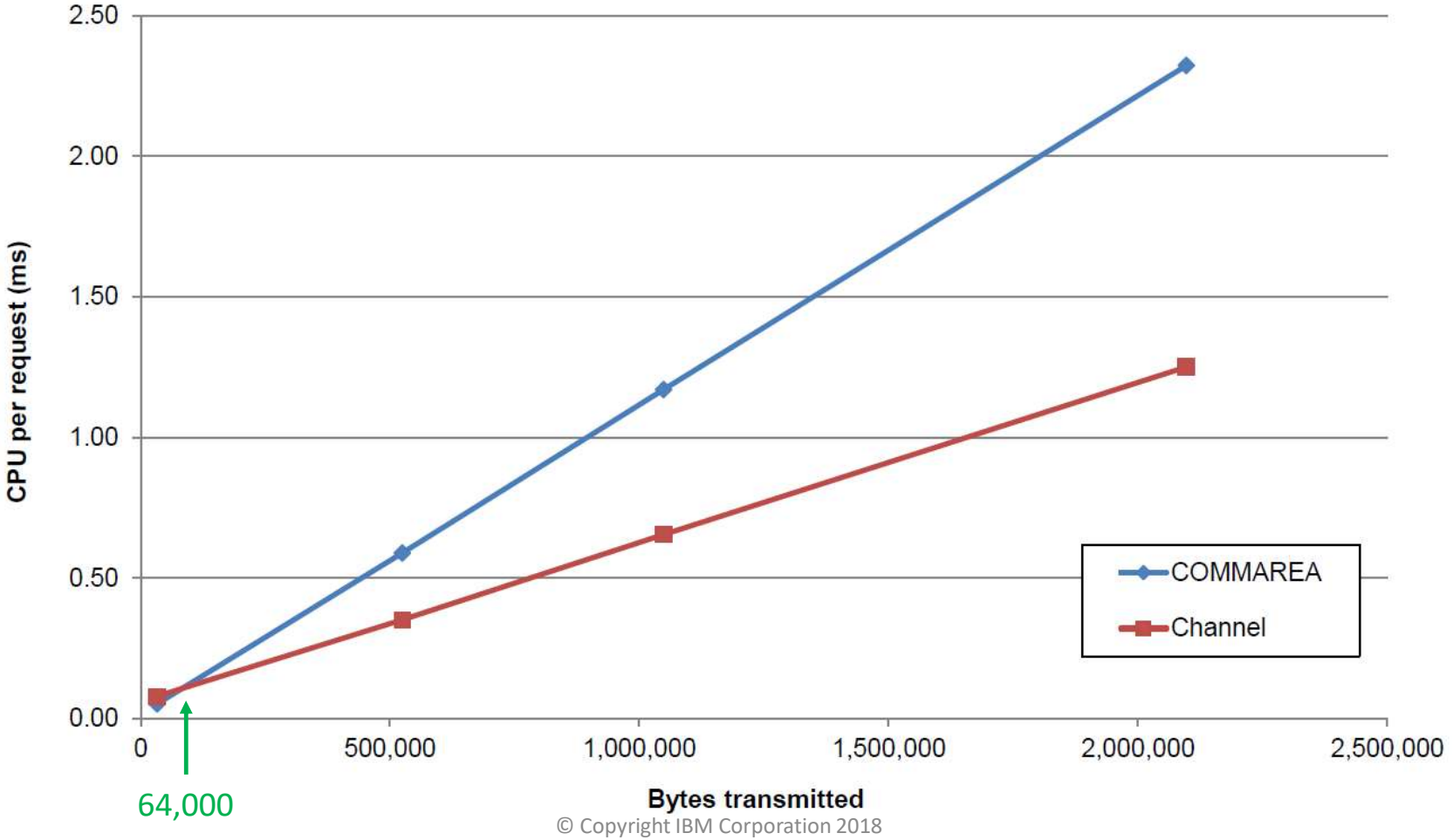


Notes

32 bytes: commarea = 0.016ms / channel = 0.057ms

32,760 bytes: commarea = 0.035ms / channel = 0.079ms

Large payload performance results



Notes

Crossover is when you need to send more than 64 kB of data (i.e. three or more calls). After that, cheaper to use channels.

Improved instrumentation

Notes

This page intentionally left blank.

Monitoring and statistics (V5.4)

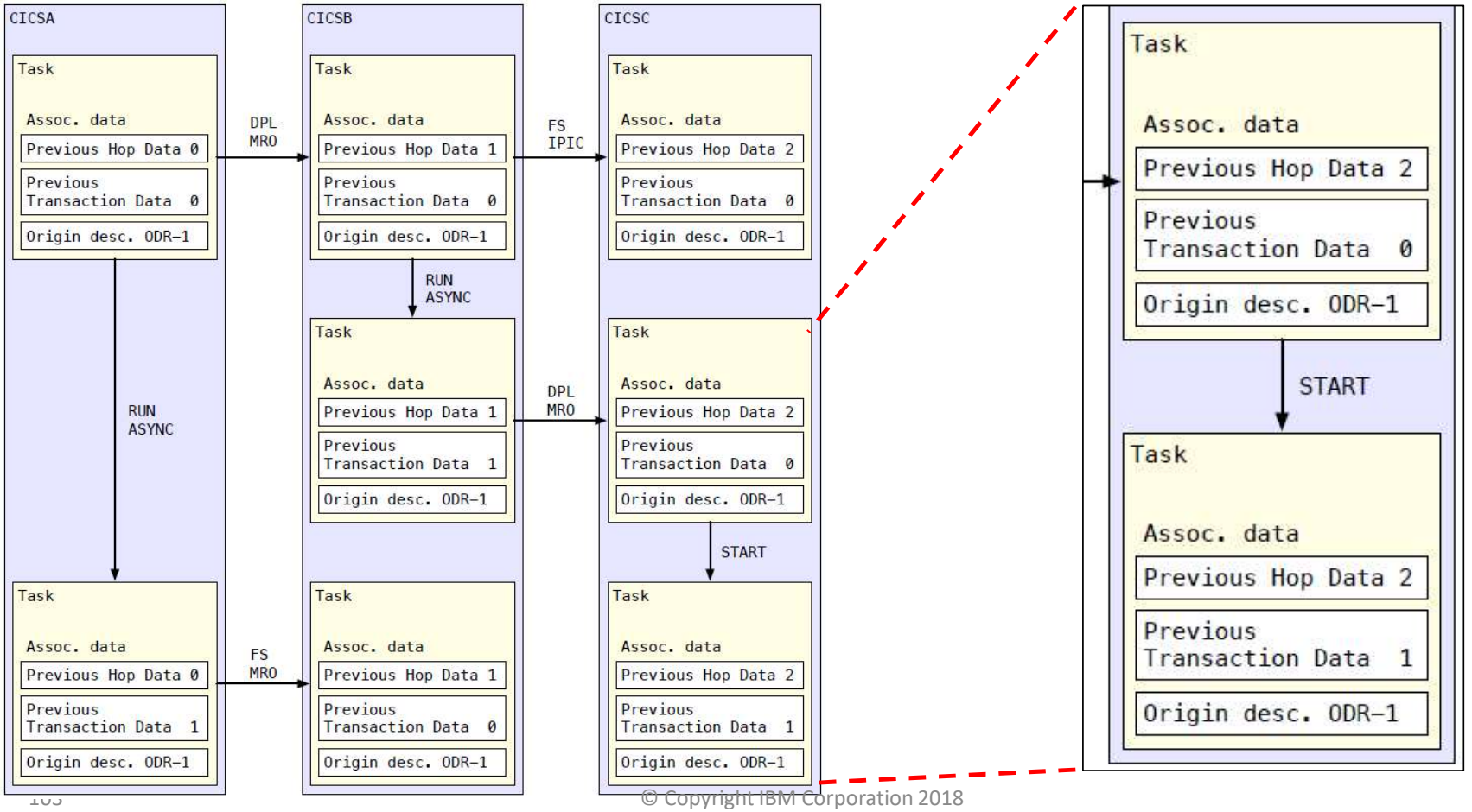
- Monitoring
  - Policy system and task rules evaluation and action counts
  - LPAR name
  - Number of asynchronous API calls
  - Enhanced transaction tracking information
- Statistics
  - Asynchronous API counts and delays
  - MQ monitor resource statistics
  - Enhancement of global and resource-level TCP/IP statistics
  - z/OS Communication Server statistics for BMS 3270 IDS

Notes

z/OS Communication Server Intrusion Detection Service (IDS) – a security facility to monitor 3270 streams and verify that data sent to the client as read-only is not being updated on return.

When both services of BMS 3270 Intrusion Detection Service by CICS and IDS by z/OS Communication Server are enabled, BMS generated 3270 data will be handled by CICS , and non-BMS 3270 data will be handled by IBM Communications Server. The advantage of enabling both is to ensure full coverage of all 3270 applications, but make use of BMS, to maximize performance and to enhance the information returned about any intrusion.

Enhanced transaction tracking (V5.4)



Notes

To support the asynchronous API, additional transaction tracking data has been added to provide a history of which task started the current task.

The previous transaction data is populated whenever an EXEC CICS RUN TRANSID or EXEC CICS START command creates a new CICS task.

Monitoring and statistics (V5.5)

- Monitoring
  - SSL cipher used for outbound web requests
  - Indicator of first message from a web client
  - Outbound web support
- Statistics
  - CICS policy rules
  - Transaction abend count
  - Node.js application

TRANSACTION - RESOURCE INFORMATION

Trans ID	Program Name	Tclass Name	Prty	Remote Name	Remote Sysid	Dyn-amic	Attach Count	Retry Count	Dynamic Local	Dynamic Remote	Remote Starts	Storage Violations	Abend Count
/FOR	DSWFORVV		1			N	461491	0	0	0	0	0	0
CADP	DFHDPLU		1			N	0	0	0	0	0	0	0
CATA	DFHZATA		255			N	0	0	0	0	0	0	0
CATD	DFHZATD		255			N	0	0	0	0	0	0	0
CATR	DFHZATR		255			N	0	0	0	0	0	0	0

Notes

Outbound SSL cipher

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

First message from a client

The SOCONNMSG field in the DFHSOCK 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

DFHWEBC group in WBSVINVK – INVOKE SERVICE

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.

Node.js application statistics

Statistics now displays installed Node.js application

Resource monitoring records (V5.5)

- New resource 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

z13 to z14 upgrade

Notes

This page intentionally left blank.

CICS PTF requirements

- All in-service releases of CICS TS support IBM z14 hardware
- Recommended APARs if using CPSM workload management:
  - PI90941 for releases of CICS TS V5
  - PI91168 for CICS TS V4.2
- No other APARs or parameter changes are required.

developer.ibm.com/answers/questions/390924/cics-ts-and-ibm-z14-compatibility-and-benefits/

Notes

Potential benefits of running CICS on Z14 :

Pervasive Encryption

Java exploitation of z14 facilities

Further improvements in cryptographic performance

Faster processing with z14

CICS HTTP workload with SSL

- Simple CICS HTTP workload
  - `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256` cipher suite
  - Persistent connections
- Reduction of 39% CPU per transaction when running on z14 comparing to z13

Notes

This page intentionally left blank.

CICS z/OS Connect EE workload

- CICS workload sending and receiving JSON documents
  - z/OS Connect EE embedded in CICS
  - `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256` cipher suite
  - Persistent connections
- Reduction of 16% CPU per transaction when running on z14 comparing to z13

Notes

This page intentionally left blank.

CICS and dataset encryption

- Simple CICS COBOL VSAM RLS workload
- Established overhead of encrypting:
  - All CICS system datasets (DFHTEMP, DFHINTRA, DFHLRQ, DFHLCD, DFHGCD)
  - All 160 VSAM application files
- Enabling encryption of all datasets on z14 increased the CPU per transaction by only 1.6%

Notes

This page intentionally left blank.

Performance publications

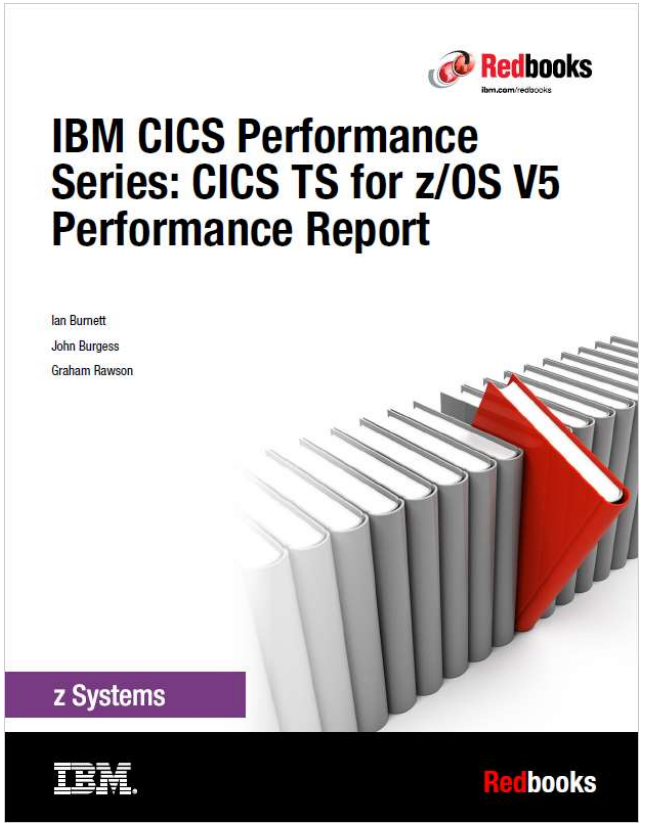
Notes

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 performance data
  - Comparisons to previous CICS releases
  - Monitoring, statistics, threadsafe enhancements
  - Performance-related SIT parameter changes

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



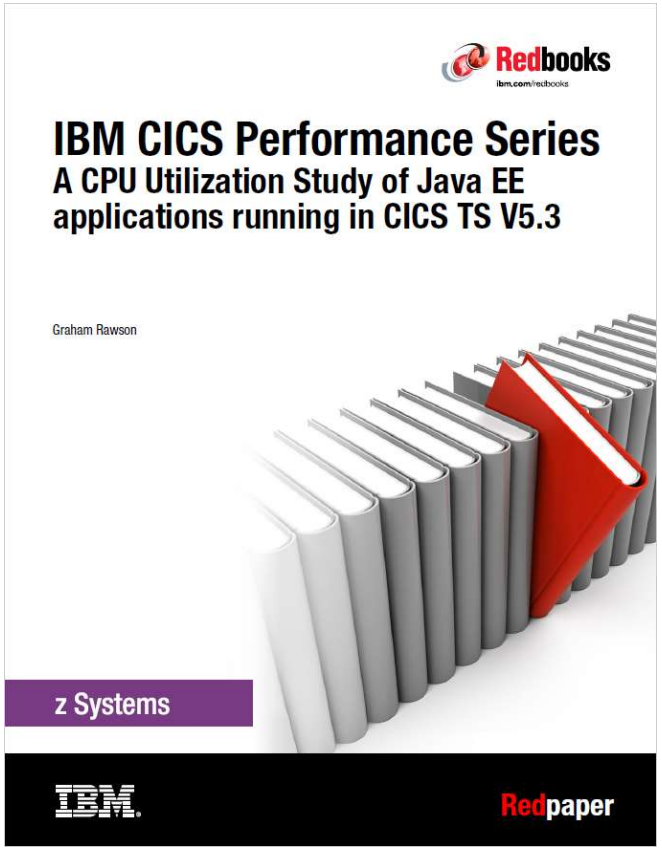
Notes

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

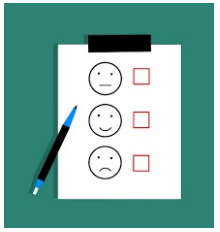


Notes

This page intentionally left blank.

We want your feedback!

- Please submit your feedback online at ….
  - http://conferences.gse.org.uk/2018/feedback/gg
- Paper feedback forms are also available from the Chair person
- This session is GG



Notices and disclaimers

- © 2018 International Business Machines Corporation. No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- **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 per 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 follows 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 about 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 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 and CICS TS for z/OS 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.