CICS Transaction Gateway for z/OS V9.0

Performance summary

Version 1.2
20 February 2014

Alan Hollingshead
alan_hollingshead@uk.ibm.com

CICS Transaction Gateway
IBM UK Laboratories
Hursley Park
Winchester
Hampshire
SO21 2JN

Table of Contents

Notices

This report is intended for Architects, Systems Programmers, Analysts and Programmers wanting to understand the performance characteristics of CICS Transaction Gateway for z/OS V9.0. The information is not intended as the specification of any programming interfaces that are provided by CICS Transaction Server for z/OS or CICS Transaction Gateway for z/OS V9.0.

It is assumed that the reader is familiar with the concepts and operation of CICS Transaction Gateway for z/OS V9.0.

References in this report to IBM products or programs do not imply that IBM intends to make these available in all countries in which IBM operates.

Information contained in this report has not been submitted to any formal IBM test and is distributed “asis”. The use of this information and the implementation of any of the techniques is the responsibility of the customer. Much depends on the ability of the customer to evaluate these data and project the results to their operational environment.

The performance data contained in this report was measured in a controlled environment and results obtained in other environments may vary significantly.

Trademarks and service marks

All statements regarding IBM plans, directions, and intent are subject to change or withdrawal without notice.

Overview

This document contains performance measurements for CICS Transaction Gateway (CICS TG) for z/OS V9.0 used in conjunction with CICS Transaction Server (CICS TS) for z/OS V5.1.

The report looks at CPU usage, including zAAP offload, for Gateway daemon and CICS address spaces, comparing EXCI and IPIC connections. It also looks at the scalability of the CICS TG and CICS TS solutions for EXCI and IPIC connections, the latter including channel payloads.

CICS Transaction Gateway and CICS Transaction Server were co-located (same LPAR and TCPIP stack), using EXCI and IPIC connectivity. The measurements were taken using the following configuration:

Hardware

- IBM System z: z10 2097-763 model E64
- 2GB of Central Storage (RAM) – unless otherwise specified
- LPAR with 3 dedicated GCPs
- LPAR with 1 zAAP – where specified
- IBM System x: x3550 M3 Intel® Xeon® 5600
- OSA-Express3 10GB Ethernet SR

Software

- CICS Transaction Gateway for z/OS V9.0
- CICS Transaction Server for z/OS V5.1
- z/OS V1R13
- IBM 31-bit SDK for z/OS Java Technology Edition, Version 7 SR4
- IBM 64-bit SDK for z/OS Java Technology Edition, Version 7 SR4

Workload

The workload simulation ran on an IBM System x machine running SUSE Linux Enterprise Server 11, using the CICS TG Java base classes to drive SYNCONRETURN ECI requests containing non-null payload data, thus avoiding null-stripping optimizations.

The CICS Transaction Server applications (one for COMMAREA requests, and one for channel requests with a single container) that received the ECI requests simply returned the payload after altering the last byte to hex '5B'.

All workloads ran at a fixed transaction rate.

Configuration

In all the scenarios:

- The CICS Transaction Gateway daemon address space and CICS Transaction Server region were co-located within a z/OS LPAR.

- Fast Local Sockets were used in the IPIC scenarios.
- TCP buffer sizes were set to 64K for both the ReceiveBufferSize and SendBufferSize.
- RMF was used to gather data about the z/OS resource usage.
- Java7 SR4 was used as performance issues in the JVM when used with CICS TG were found with SR2 and SR3.
- CTGSTART_OPTS for 31-bit Gateway daemon included the JVM system property override:
  - shareclasses=none
- CTGSTART_OPTS for 64-bit Gateway daemon included the JVM system property overrides:
  - shareclasses=none
  - comparessedrefs

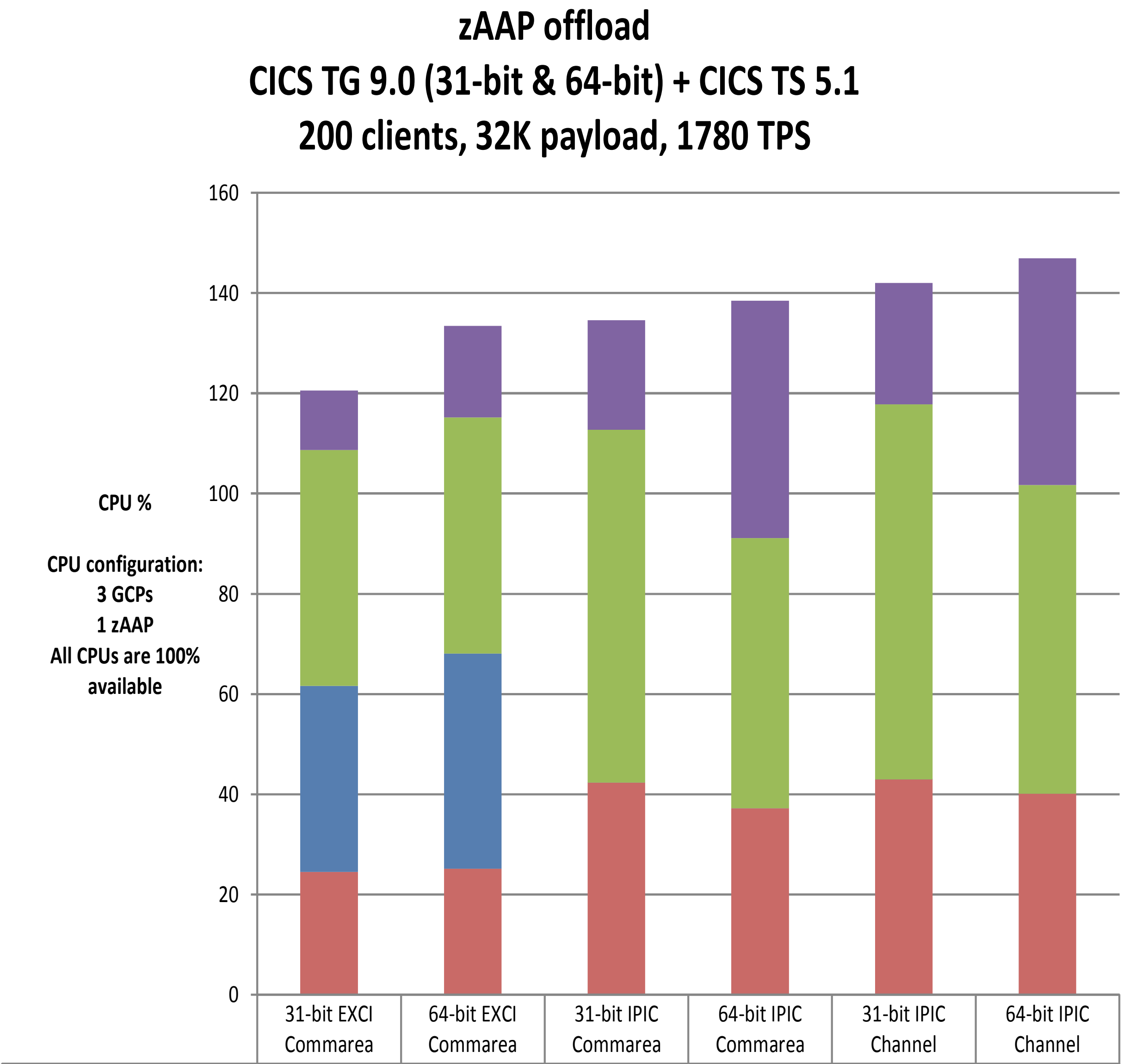
Terminology

GCP / CPU	- IBM System z General Purpose CPU
ZAAP	- IBM System z Application Assist Processor
CPU %	- Percentage of CPU time used by transactions running on general purpose processors
Cost per transaction (ms)	- CPU usage per transaction, in milliseconds
TPS	- Number of Transactions Per Second
CICS TG	- IBM CICS Transaction Gateway for z/OS
CICS TS	- IBM CICS Transaction Server for z/OS
IPIC	- Internet Protocol (IP) interconnectivity
RMF	- Resource Measurement Facility

Scenario 1: zAAP offload with EXCI and IPIC connections

This scenario compared EXCI and IPIC (COMMAREA and channel) using the same payload size at an approximate transaction rate of 1780 TPS with both a 31-bit and 64-bit Gateway daemon.

A single zAAP was available to offload eligible workload.



	31-bit EXCI Commarea	64-bit EXCI Commarea	31-bit IPIC Commarea	64-bit IPIC Commarea	31-bit IPIC Channel	64-bit IPIC Channel
CICS TG zAAP (potential) %	11.82	18.25	21.80	47.33	24.26	45.25
CICS TG zAAP (actual) %	47.12	47.10	70.41	53.94	74.80	61.62
CICS TG non-zAAP eligible %	37.12	42.92	0.02	0.03	0.02	0.00
CICS TS non-zAAP eligible %	24.49	25.18	42.32	37.17	42.98	40.10

Illustration 1: "zAAP offload comparison"

Observations

Illustration 1: “zAAP offload comparison” shows that the total GCP load (red and blue bars combined) is lower for IPIC when zAAP offload is available. The CICS TG non-zAAP CPU load (blue) is so low for IPIC that it is invisible on the graph.

The single zAAP was unable to satisfy all the eligible workload; the full potential offload is therefore represented by both green and purple areas on the graph.

31-bit Gateway daemon:

Protocol / Payload	**CICS TG CPU load zAAP eligible (%)**	**CICS TS+TG CPU load zAAP eligible (%)**	**GCP load %**	**Potential GCP saving with IPIC vs EXCI**
EXCI / COMMAREA	61.36%	48.89%	51.11%	n/a
IPIC / COMMAREA	99.98%	68.54%	31.46%	**19.65%**
IPIC / Channel	99.98%	69.74%	30.26%	**20.85%**

Table 1: “zAAP offload summary for 31-bit Gateway daemon”

64-bit Gateway daemon:

Protocol / Payload	**CICS TG CPU load zAAP eligible (%)**	**CICS TS+TG CPU load zAAP eligible (%)**	**GCP load %**	**Potential GCP saving with IPIC vs EXCI**
EXCI / COMMAREA	60.36%	48.97%	51.03%	n/a
IPIC / COMMAREA	99.97%	73.15%	26.85%	**24.18%**
IPIC / Channel	100.00%	72.72%	27.28%	**23.75%**

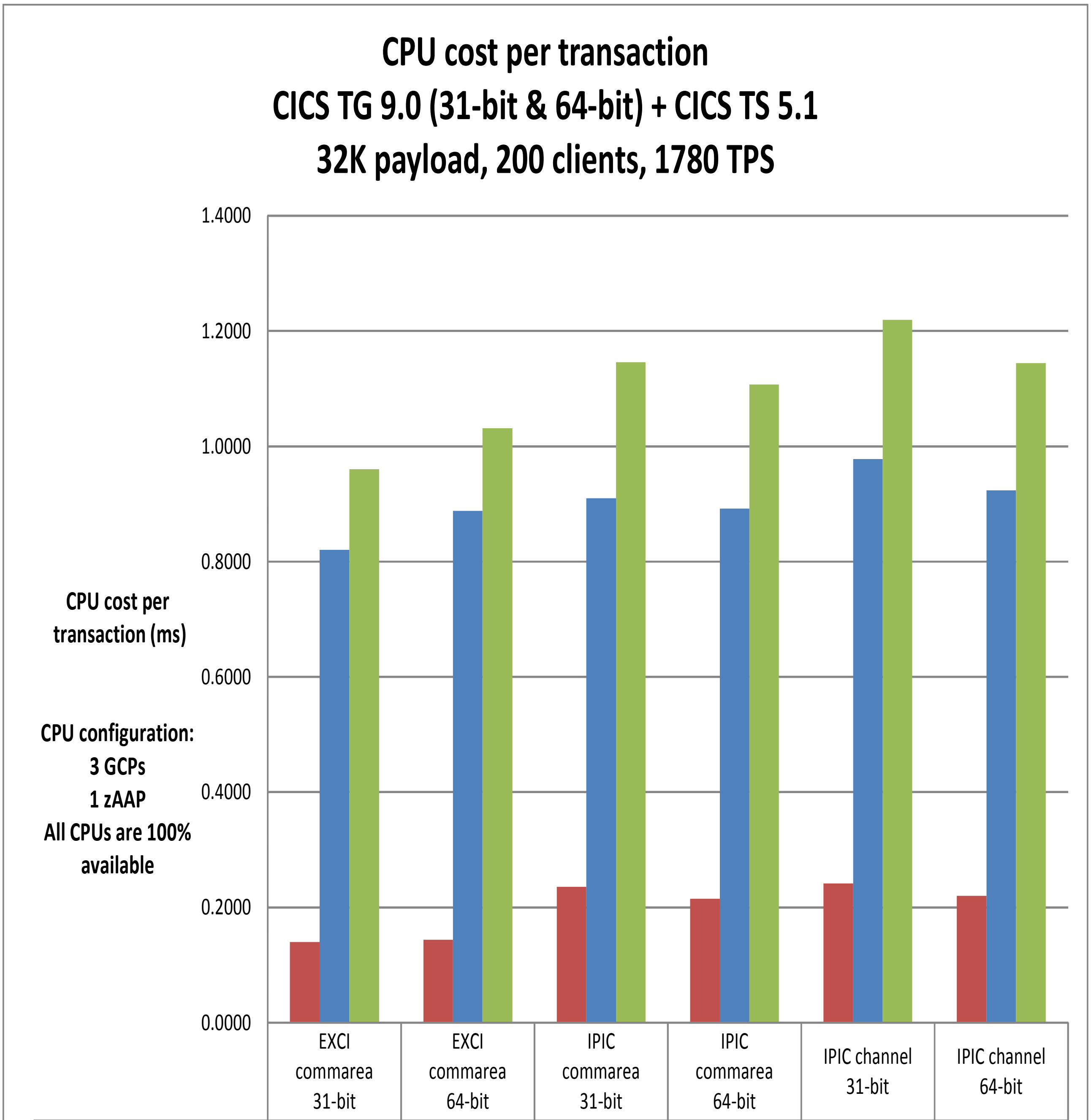
Table 2: “zAAP offload summary for 64-bit Gateway daemon”

Table 1: “zAAP offload summary for 31-bit Gateway daemon” and Table 2: “zAAP offload summary for 64-bit Gateway daemon” show the zAAP eligibility for CPU loads in the CICS TG address space, and in context with CICS TS address space.

The observed overall reduction in general purpose CPU load makes IPIC connectivity an attractive option for zAAP owners migrating their workload from EXCI to IPIC.

CICS TG configuration for these measurements:

	REGION	MEMLIMIT	Heap (Xmx)
EXCI 31-bit	1000M	n/a	500M
IPIC 31-bit	1000M	n/a	500M
IPIC 64-bit	300M	4M	2048M



	EXCI commarea 31-bit	EXCI commarea 64-bit	IPIC commarea 31-bit	IPIC commarea 64-bit	IPIC channel 31-bit	IPIC channel 64-bit
CICS TS 5.1 CPU cost per Txn	0.1401	0.1440	0.2358	0.2150	0.2416	0.2204
CICS TG 9.0 CPU cost per Txn	0.8204	0.8876	0.9099	0.8919	0.9780	0.9238
Total CPU cost per Txn	0.9605	1.0316	1.1457	1.1069	1.2196	1.1442

Illustration 2: "CPU cost per transaction for Scenario 1"

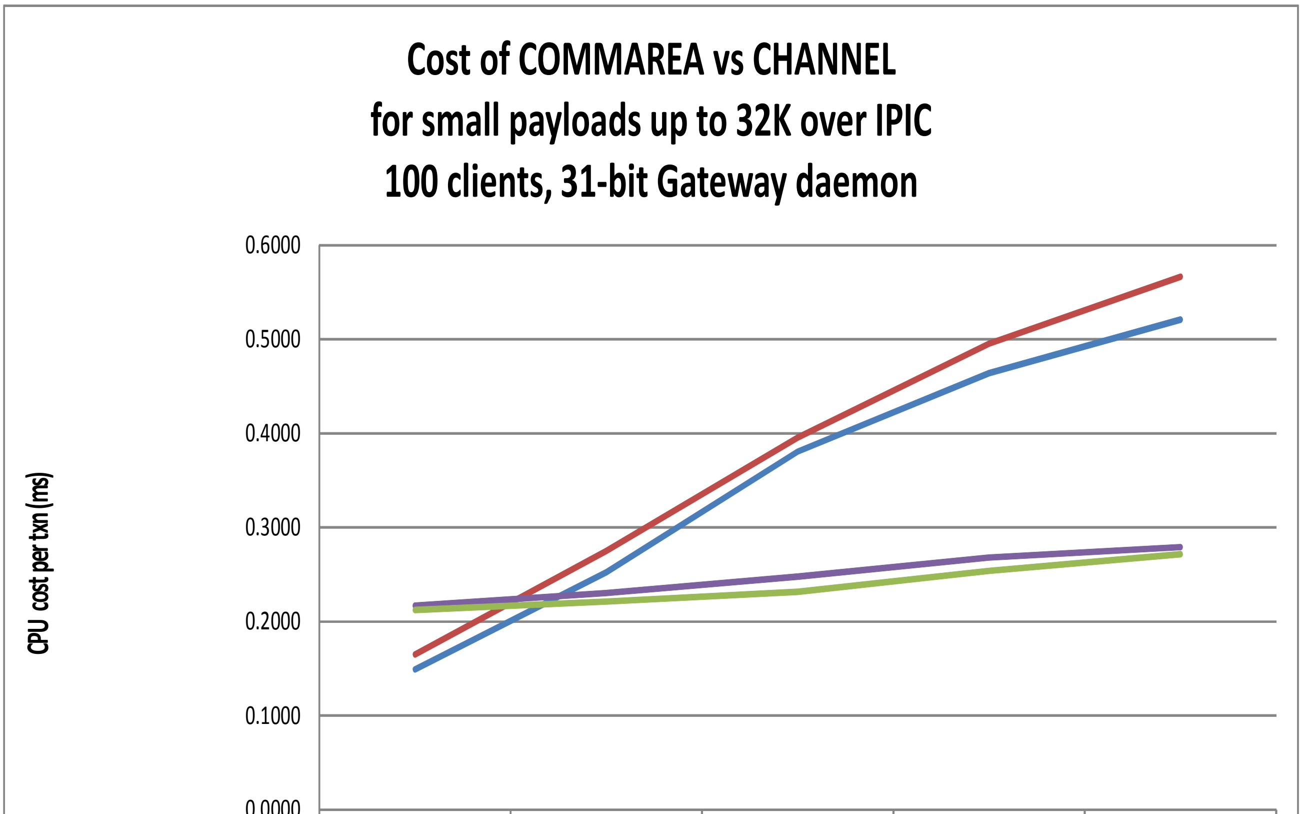
Illustration 2: "CPU cost per transaction for Scenario 1" uses the same measurements as for Illustration 1: "zAAP offload comparison" this time showing the CPU costs per transaction comparing EXCI with IPIC commareas and channels, for both a 31-bit and 64-bit Gateway daemon.

Observations
Although the total path length is slightly longer for both CICS TS and CICS TG when using IPIC, the offload potential is such that GCP costs are lowered.

The transactions per second, 1780 TPS, is approximately the same on both IPIC and EXCI, for both the 31-bit or 64-bit Gateway daemon, regardless of the protocol and payload type in use.

Scenario 2: Comparing the cost of COMMAREA vs channel for small payloads up to 32K

This scenario compared the CICS TG CPU cost per transaction for COMMAREAs and channels using payloads up to 32K. The scenario was run using 31-bit and 64-bit Gateway daemons.

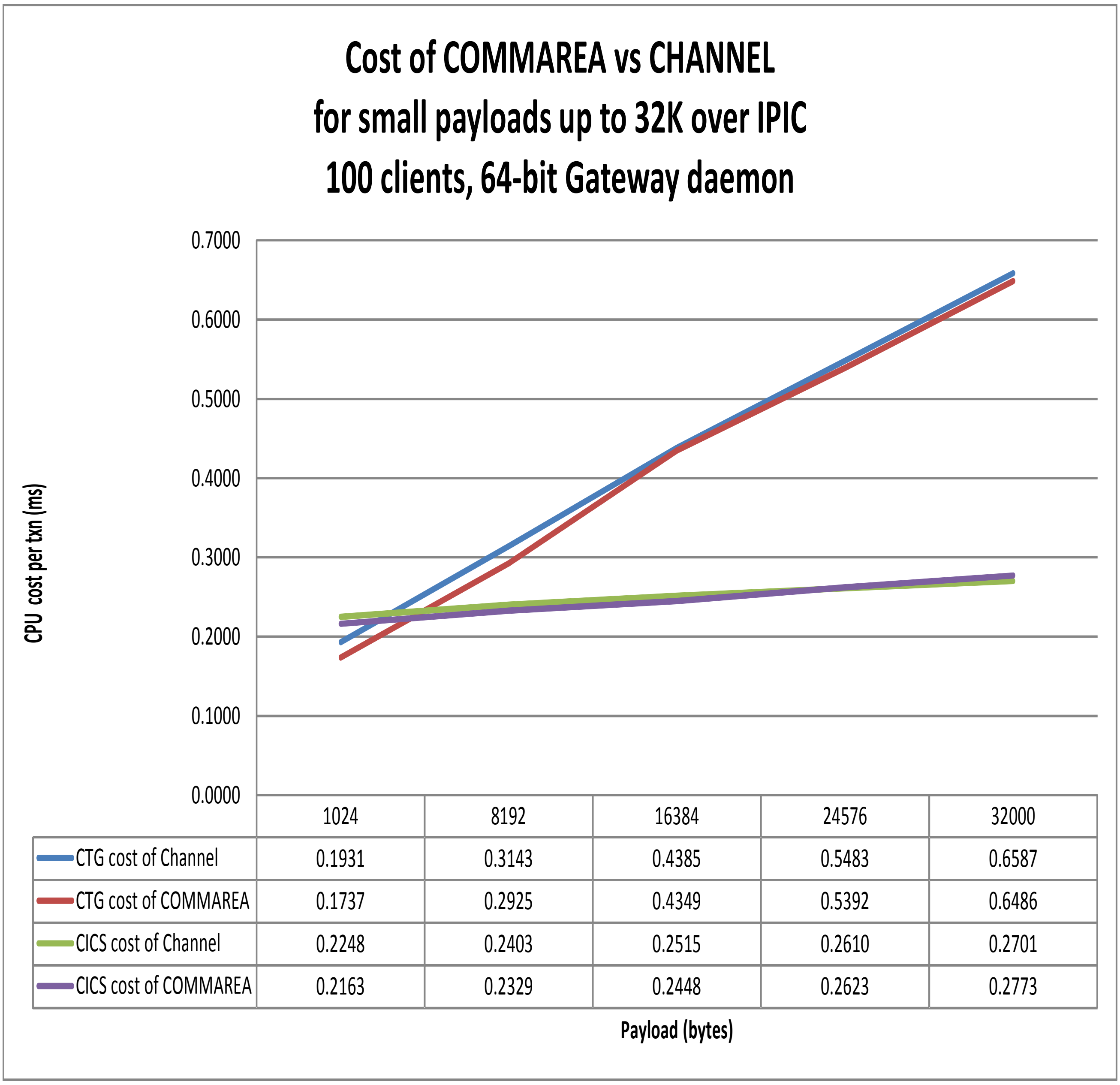


	1024	8192	16384	24576	32000
CICS TG cost of Channel	0.1653	0.2752	0.3956	0.4954	0.5665
CICS TG cost of COMMAREA	0.1492	0.2526	0.3809	0.4644	0.5210
CICS cost of Channel	0.2172	0.2305	0.2479	0.2682	0.2794
CICS cost of COMMAREA	0.2123	0.2214	0.2317	0.2541	0.2716

Payload (bytes)

Illustration 3: "COMMAREA vs channel cost of CPU per transaction – 31-bit Gateway daemon"

Illustration 3: "COMMAREA vs channel cost of CPU per transaction – 31-bit Gateway daemon" demonstrates how the CPU cost per transaction compares for COMMAREAs and channels for varying payload sizes up to 32000 bytes using a 31-bit Gateway daemon.



	1024	8192	16384	24576	32000
CTG cost of Channel	0.1931	0.3143	0.4385	0.5483	0.6587
CTG cost of COMMAREA	0.1737	0.2925	0.4349	0.5392	0.6486
CICS cost of Channel	0.2248	0.2403	0.2515	0.2610	0.2701
CICS cost of COMMAREA	0.2163	0.2329	0.2448	0.2623	0.2773

Illustration 4: “COMMAREA vs channel cost of CPU per transaction – 64-bit Gateway daemon” demonstrates the scenario using a 64-bit Gateway daemon.

Observations

The results in Illustration 3: “COMMAREA vs channel cost of CPU per transaction – 31-bit Gateway daemon” and Illustration 4: “COMMAREA vs channel cost of CPU per transaction – 64-bit Gateway daemon” show that the performance costs for channels versus COMMAREAs for small workloads up to 32K are approximately the same on both the 31-bit and 64-bit Gateway daemon.

When writing new applications that will be using the IPIC protocol into CICS TS, the benefits of using channels rather than COMMAREAs should be strongly considered as they provide greater flexibility and capabilities (see the CICS TS Information Center for more details).

CICS TG configuration for these measurements:

	REGION	MEMLIMIT	Heap (Xmx)
IPIC 31-bit	1000M	n/a	500M
IPIC 64-bit	600M	20M	2048M

The measurements for this scenario were taken whilst 10GB of RAM was allocated to the z10.

No zAAP was used for this set of measurements.

Scenario 3: IPIC clients scaling with small channel payload

This scenario compared the TPS and the CPU cost per transaction of the 31-bit and 64-bit Gateway daemons running a 4K channel workload over IPIC, varying the number of clients.

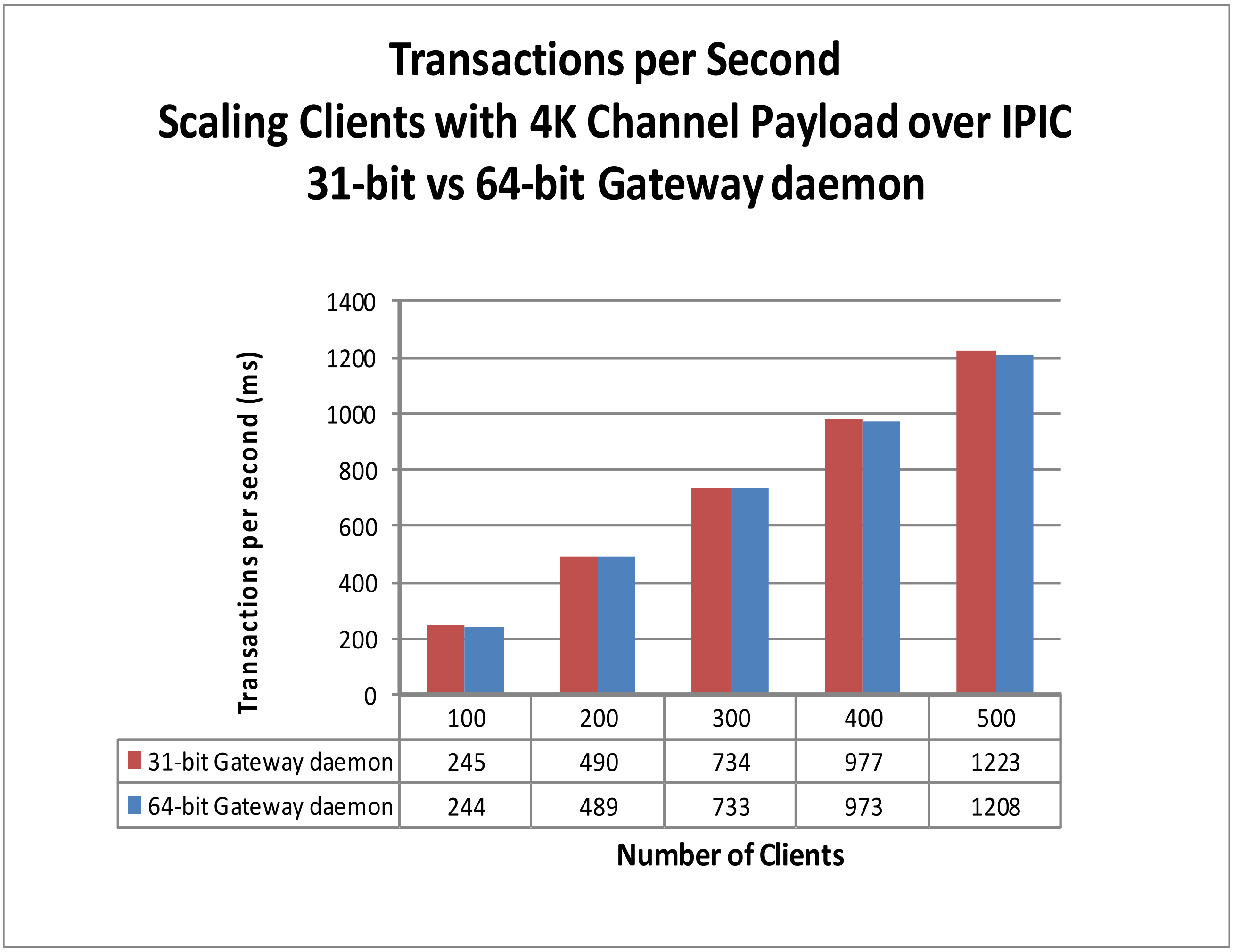


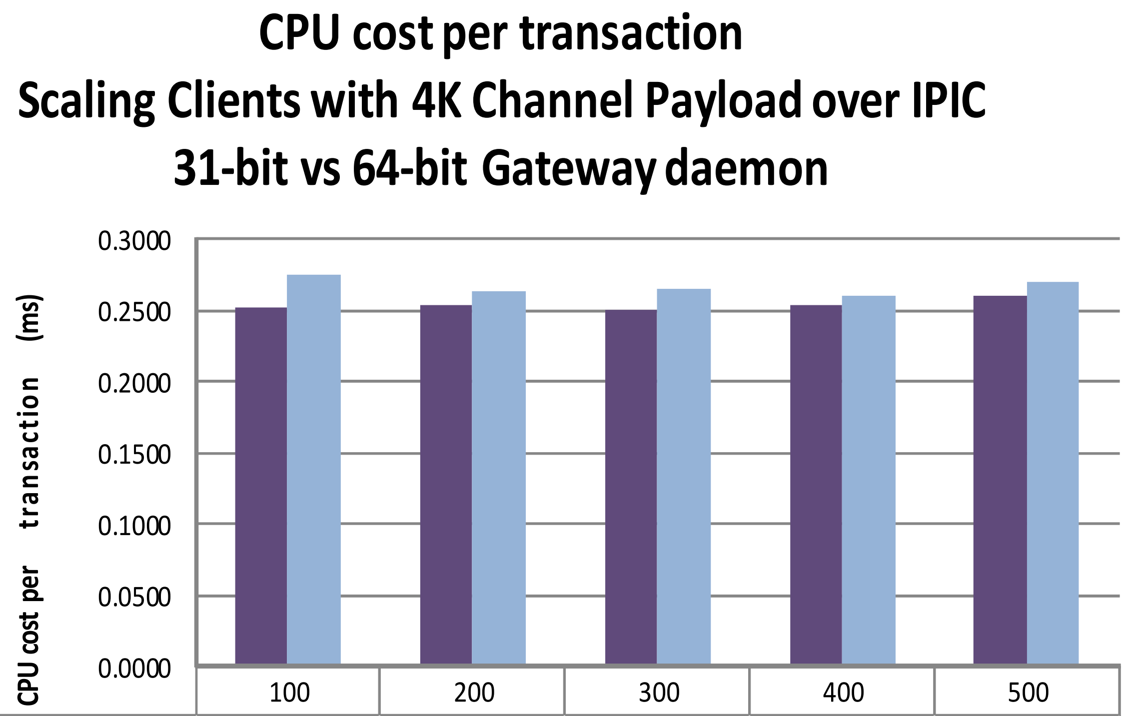
Illustration 5: "IPIC scaling clients TPS with 4K payload"

Observations

Illustration 5: "IPIC scaling clients TPS with 4K payload" demonstrates how the TPS increases linearly in line with the number of clients running simultaneously.

The graph also shows that the TPS achieved is almost identical whether using the 31-bit or 64-bit Gateway daemon.

Having established that the TPS is approximately the same for the 31-bit Gateway daemon and the 64-bit Gateway daemon, the CPU cost per transaction was observed.



	100	200	300	400	500
31-bit Gateway daemon	0.2515	0.2540	0.2502	0.2538	0.2595
64-bit Gateway daemon	0.2745	0.2629	0.2648	0.2608	0.2696

Illustration 6: "Scaling clients comparing 31-bit & 64-bit Gateway daemon"

Observations

Using the same measurements as shown in Illustration 5: "IPIC scaling clients TPS with 4K payload", the graph above in Illustration 6: "Scaling clients comparing 31-bit & 64-bit Gateway daemon" shows how the CPU cost per transaction is maintained at a fairly consistent level regardless of the number of clients running simultaneously. This indicates that the pathlength through the Gateway daemon and CICS Transaction Server is approximately the same.

The 64-bit Gateway daemon for small payloads, such as 4K channels, is seen to be slightly more expensive (between 2.7% and 9%) than the same workload running on the 31-bit Gateway daemon.

CICS TG configuration for these measurements:

	REGION	MEMLIMIT	Heap (Xmx)
31-bit Gateway	1000M	n/a	500M
64-bit Gateway	600M	12G	2048M

The measurements for this scenario were taken whilst 10GB of RAM was allocated to the z10.

No zAAP was used for this set of measurements.

Scenario 4: IPIC clients scaling with 32K channel payload

This scenario compared the TPS and the CPU cost per transaction of the 31-bit and 64-bit Gateway daemons running a 32K channel workload over IPIC, varying the number of clients.

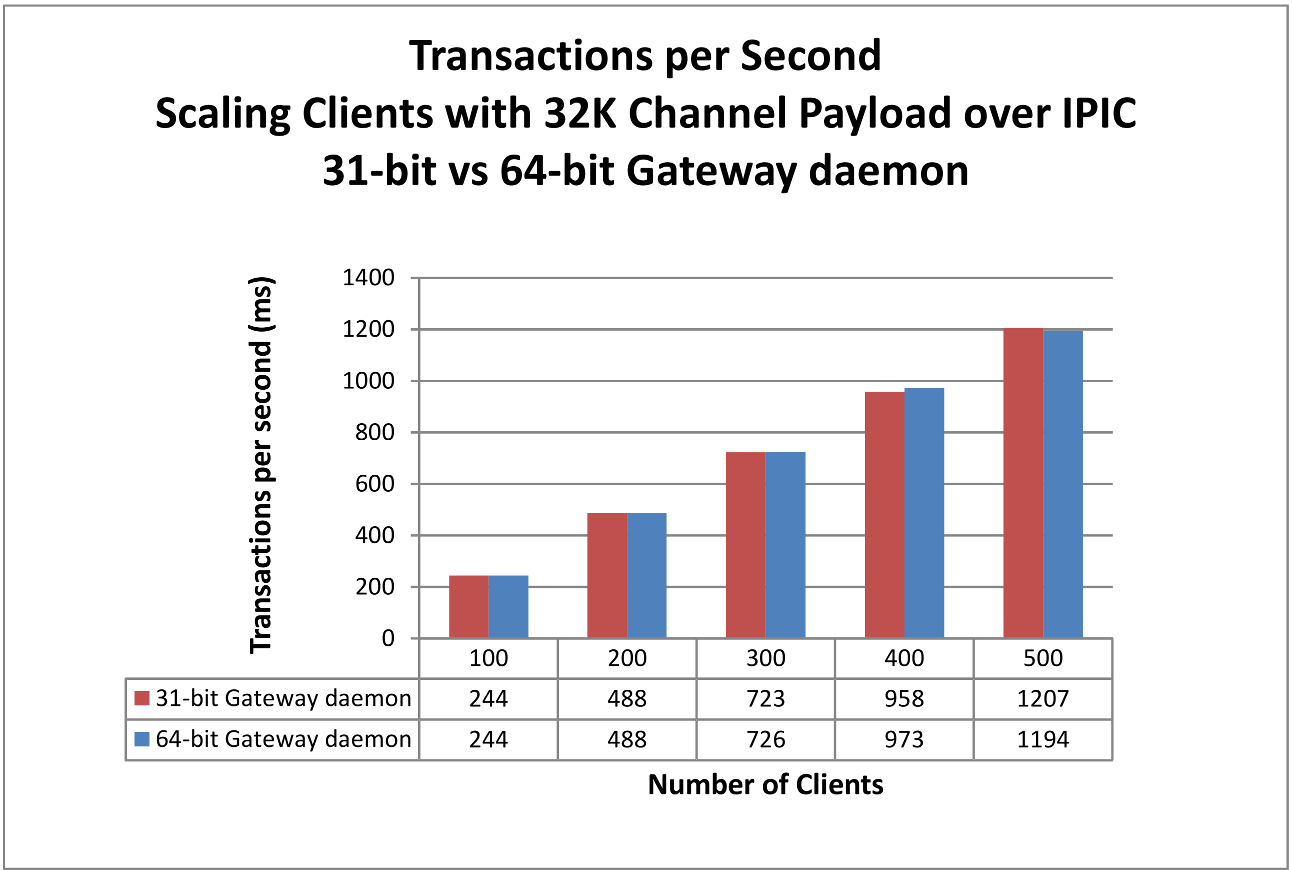


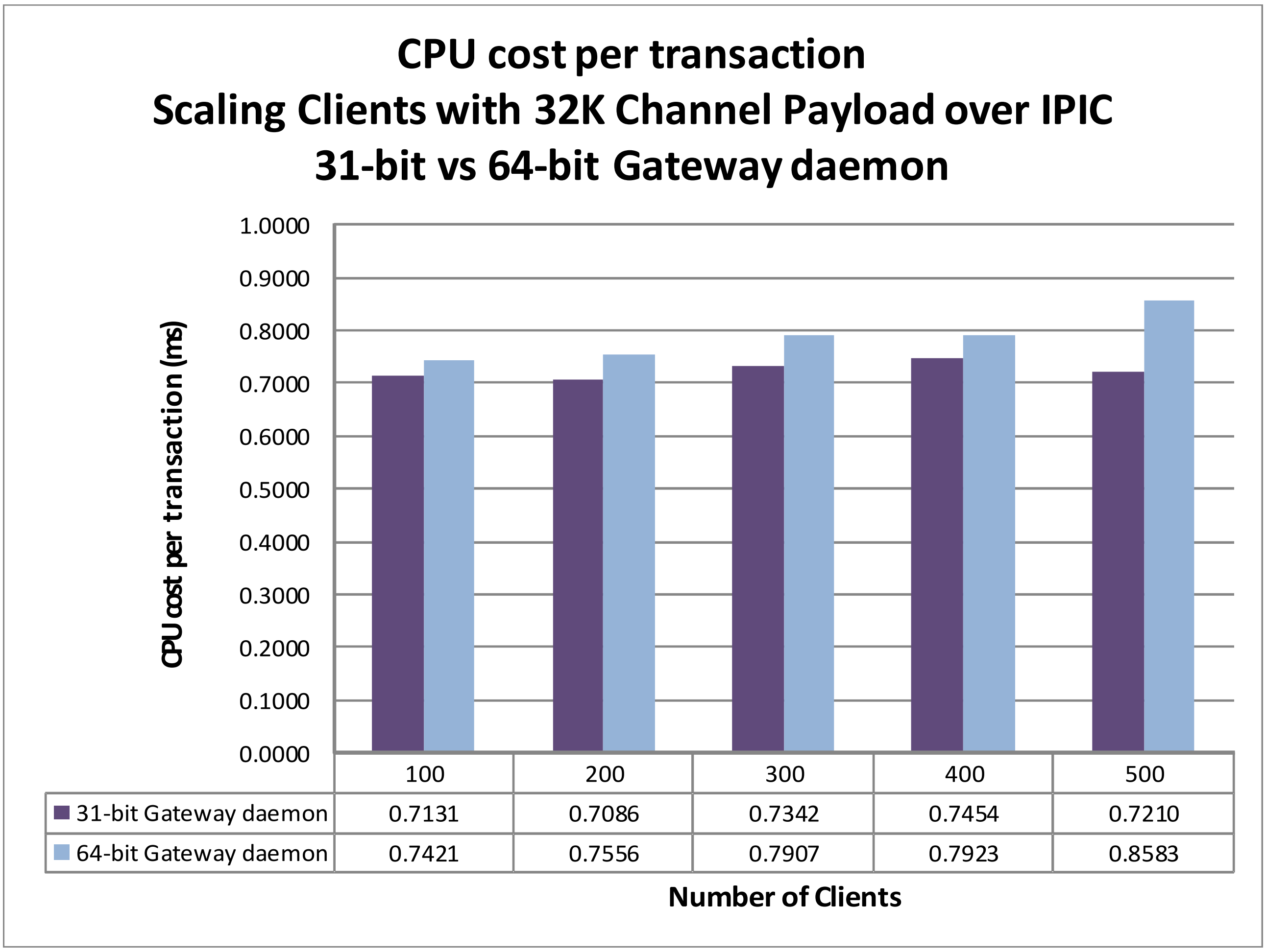
Illustration 7: "Scaling clients comparing 31-bit & 64-bit Gateway daemon"

Observations

Illustration 7: "Scaling clients comparing 31-bit & 64-bit Gateway daemon" demonstrates how the TPS increases linearly in line with the number of clients running simultaneously.

The graph also shows that the TPS is almost identical whether using the 31-bit or 64-bit Gateway daemon.

Having established that the TPS is approximately the same for the 31-bit Gateway daemon and the 64-bit Gateway daemon, the CPU cost per transaction was observed.



	100	200	300	400	500
31-bit Gateway daemon	0.7131	0.7086	0.7342	0.7454	0.7210
64-bit Gateway daemon	0.7421	0.7556	0.7907	0.7923	0.8583

Illustration 8: "IPIC scaling clients TPS with 32K payload"

Observations

Using the same measurements as shown in Illustration 7: "Scaling clients comparing 31-bit & 64-bit Gateway daemon", the graph above in Illustration 8: "IPIC scaling clients TPS with 32K payload" demonstrates that a fairly consistent CPU cost per transaction is maintained regardless of the number of clients running simultaneously. This indicates that the pathlength through the Gateway daemon and CICS transaction server is approximately the same.

The 64-bit Gateway daemon for 32K payloads, is seen to be more expensive (between 4% and 19%) than the same workload running on the 31-bit Gateway daemon.

CICS TG configuration for these measurements:

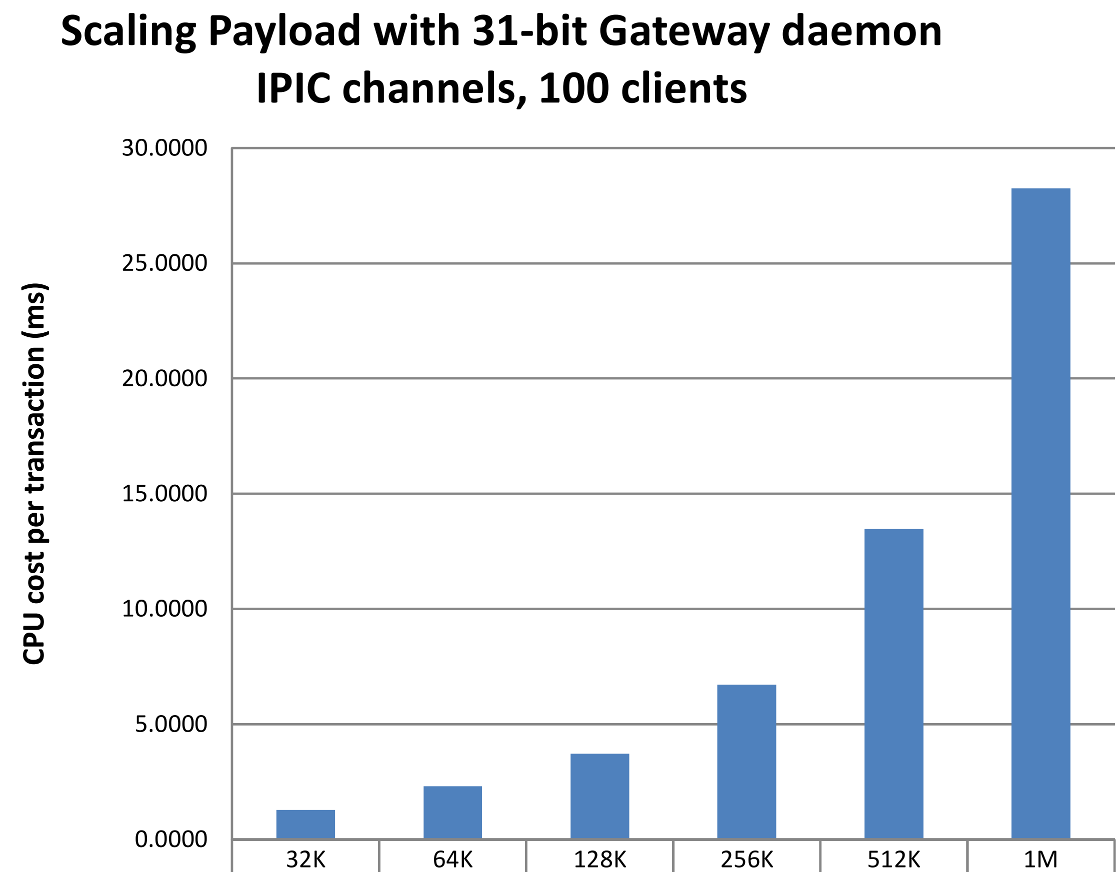
	REGION	MEMLIMIT	Heap (Xmx)
31-bit Gateway	1000M	n/a	500M
64-bit Gateway	600M	12G	2048M

The measurements for this scenario were taken whilst 10GB of RAM was allocated to the z10.

No zAAP was used for this set of measurements.

Scenario 5: IPIC payload scaling with 31-bit Gateway daemon

This scenario monitored the CPU cost per transaction of the 31-bit Gateway daemon whilst the channel payload doubled in size over IPIC, using 100 clients.



	32K	64K	128K	256K	512K	1M
CICS TG IPIC Channel 31-bit	1.2743	2.3075	3.7198	6.7124	13.4555	28.2343

Illustration 9: "IPIC payload doubling in size"

Observations

Illustration 9: "IPIC payload doubling in size" demonstrates that as the payload doubles in size (32K, 64K, 128K, etc), the CPU cost per transaction also doubles. This shows that even with large payloads the 31-bit Gateway daemon scaled proportionally regardless of the payload size.

A single zAAP was enabled whilst these measurements were taken.

CICS TG configuration for these measurements:

	REGION	MEMLIMIT	Heap (Xmx)
IPIC 31-bit	800M	n/a	500M

Scenario 6: IPIC payload scaling with 64-bit Gateway daemon

This scenario monitored the CPU cost per transaction of the 64-bit Gateway daemon whilst the channel payload increased in increments of 1MG in size over IPIC, using 100 clients.

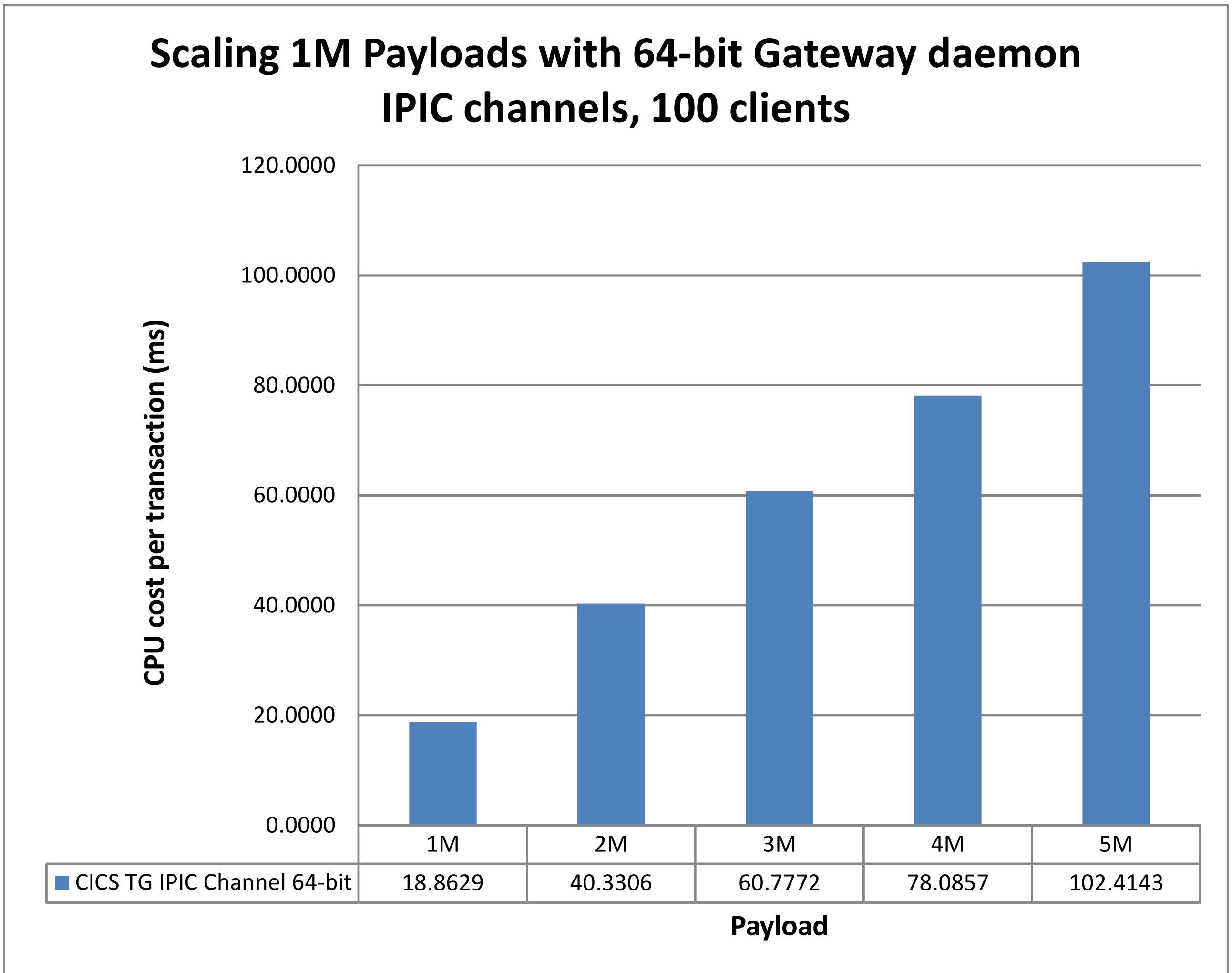


Illustration 10: "IPIC payloads increasing by 1M"

Observations

Illustration 10: "IPIC payloads increasing by 1M" demonstrates that as the payload increases in a linear fashion (1M, 2M, 3M, etc), the CPU cost per transaction also increases linearly. This shows that even with large payloads the 64-bit Gateway daemon scaled proportionally regardless of the payload size.

No zAAP was used for this set of measurements.

CICS TG configuration for these measurements:

	REGION	MEMLIMIT	Heap (Xmx)
IPIC 64-bit	300M	4M	2048M

Conclusions

Customers should consider the following:

- When using IPIC, zAAP/zIIPs can provide large benefits by offloading eligible work, thus potentially reducing the cost of running workloads.
- Channels (rather than COMMAREAs) provide greater benefits and flexibility, and should be considered when writing new applications.
- The 31-bit Gateway daemon is ideal for small payloads and small numbers of clients, and will typically offer a greater CPU cost per transaction compared to a 64-bit Gateway daemon running the same workload.
- The 64-bit Gateway daemon offers good scalability for both large payloads and large numbers of clients.
- Both the 31-bit and 64-bit Gateway daemons performed at a similar TPS when running the same workloads.

Notes:

1. Analysis of other payload sizes, at other transaction rates, have not been completed at this time, so there is no guarantee that equivalent observations will be seen in other configurations.
2. Due to the effects on system performance of machine hardware, levels of software configuration and payload, equivalent observations might not be seen on other systems.