CICS Transaction Gateway for z/OS V8.0

Performance summary for IPIC connections

Version 1.0

Rob Jones
rcjones@uk.ibm.com

CICS Transaction Gateway Development
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 V8.0. 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 V8.0.

It is assumed that the reader is familiar with the concepts and operation of CICS Transaction Gateway for z/OS V8.0.

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.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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.

Overview

This document contains performance measurements for CICS Transaction Gateway for z/OS V8.0 and CICS Transaction Server for z/OS V4.1. Specifically, the use of IPIC connections between a co-located (same LPAR + TCPIP stack) Gateway daemon and CICS server is illustrated for ECI requests, including both COMMAREA and CHANNEL payloads.

The measurements were taken using the following configuration:

Hardware

- Z10 2097-763 model E64
- LPAR with 3 dedicated Cps
- Intel(R) Xeon(R) CPU E5430 @ 2.66GHz (8 cores)
- 1 Gbps LAN

Software

- CICS Transaction Server for z/OS V4.1
- CICS Transaction Gateway for z/OS V8.0
- z/OS V1R12
- IBM 31-bit SDK for z/OS Java Technology Edition v6 SR8
- IBM TCPIP configured with 64KB default send/receive buffers
- Red Hat Enterprise Linux Server release 5.2

Workload

The workload simulation runs on a Intel Xeon multi-processor machine running Red Hat Enterprise Linux, using the CICS TG Java base classes to drive SYNCONRETURN ECI requests containing non-null payload data, thus avoiding null-stripping optimizations.

Terminology

"CPU Load %" can be interpreted as *CPU usage* (i.e. % of 1CP).
"CICS TS" refers to IBM CICS Transaction Server for z/OS.
"CICS TG" refers to IBM CICS Transaction Gateway for z/OS.
"IPIC" refers to Internet Protocol (IP) interconnectivity.
"TPS" refers to the number of Transactions Per Second.

IPIC Workload scaling with COMMAREA

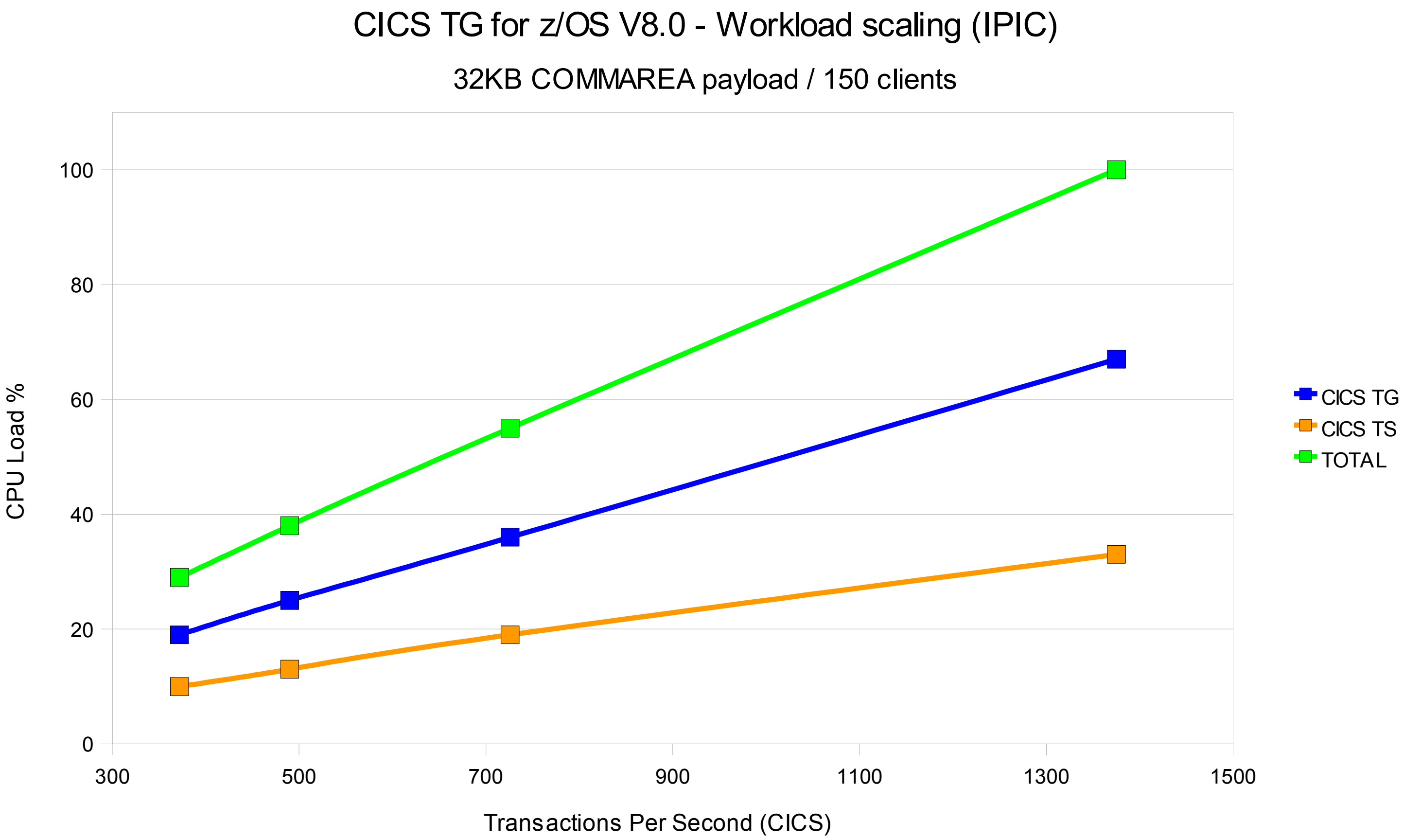

Illustration 1: 32KB COMMAREA / 150 Clients

Notes

In this scenario, the COMMAREA payload and number of clients remains constant (32KB, 150 clients) whilst the the frequency of each application request increases, driving a higher transaction rate. CPU usage of CICS TG and CICS TS increases proportionally to the transaction rate.

This could correspond to a relatively stable number of users hitting a daily peak in activity.

IPIC payload scaling with COMMAREA

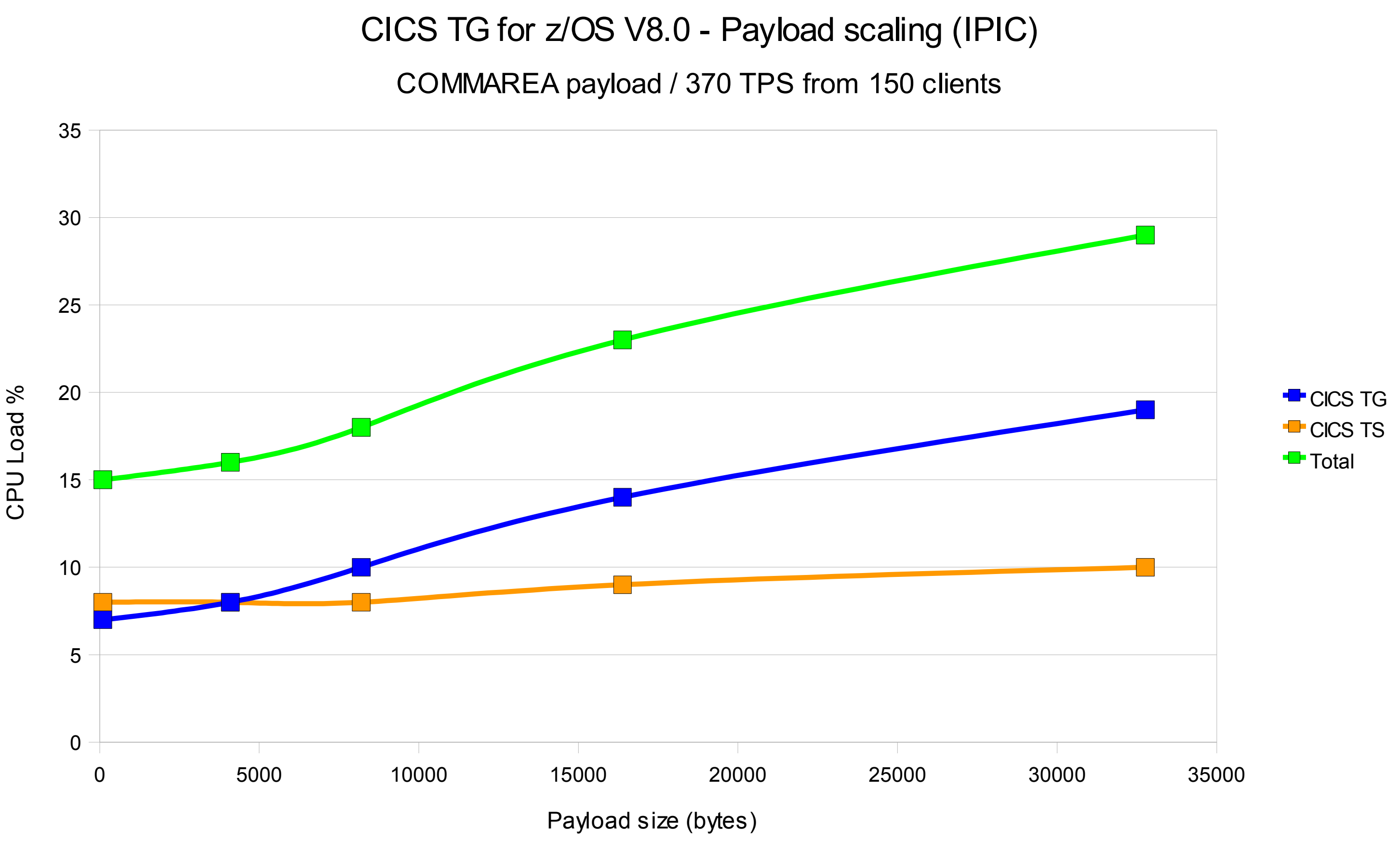

Illustration 2: COMMAREA payload / 370 TPS from 150 clients

Notes

In this scenario, a stable rate of requests (370 TPS) is maintained whilst the size of the COMMAREA payload sent/received by the client application increases from 100 bytes to 32KB.

This shows that IPIC connections scale well for ECI COMMAREA requests as payload sizes increase. CPU usage of CICS TG and CICS TS is proportional to average payload size of the workload.

IPIC Workload scaling with CHANNEL

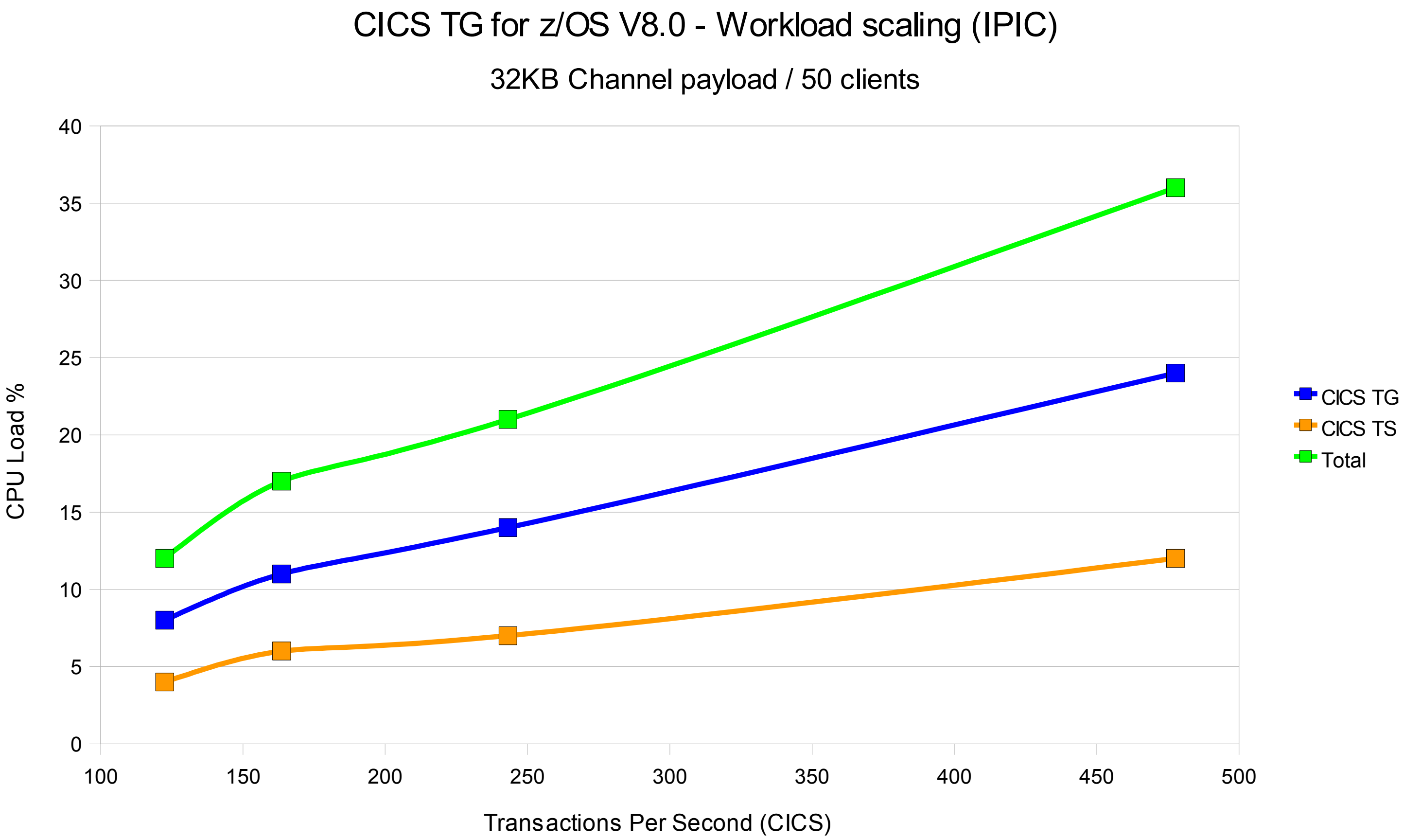

Illustration 3: 32KB Channel payload / 50 clients

Notes

In this scenario, the CHANNEL payload and number of clients remains constant (32KB, 50 clients) whilst the the frequency of each application request increases, driving a higher transaction rate.

This could correspond to a relatively stable number of users hitting a daily peak in activity, and shows that CPU load follows a similar CPU load pattern to COMMAREA workloads. CPU usage of CICS TG and CICS TS increases proportionally to the transaction rate.

The range of transaction rates here (and hence CPU load) are lower compared with the COMMAREA scenario, since the number of clients were 50 rather than 150.

IPIC payload scaling with CHANNEL

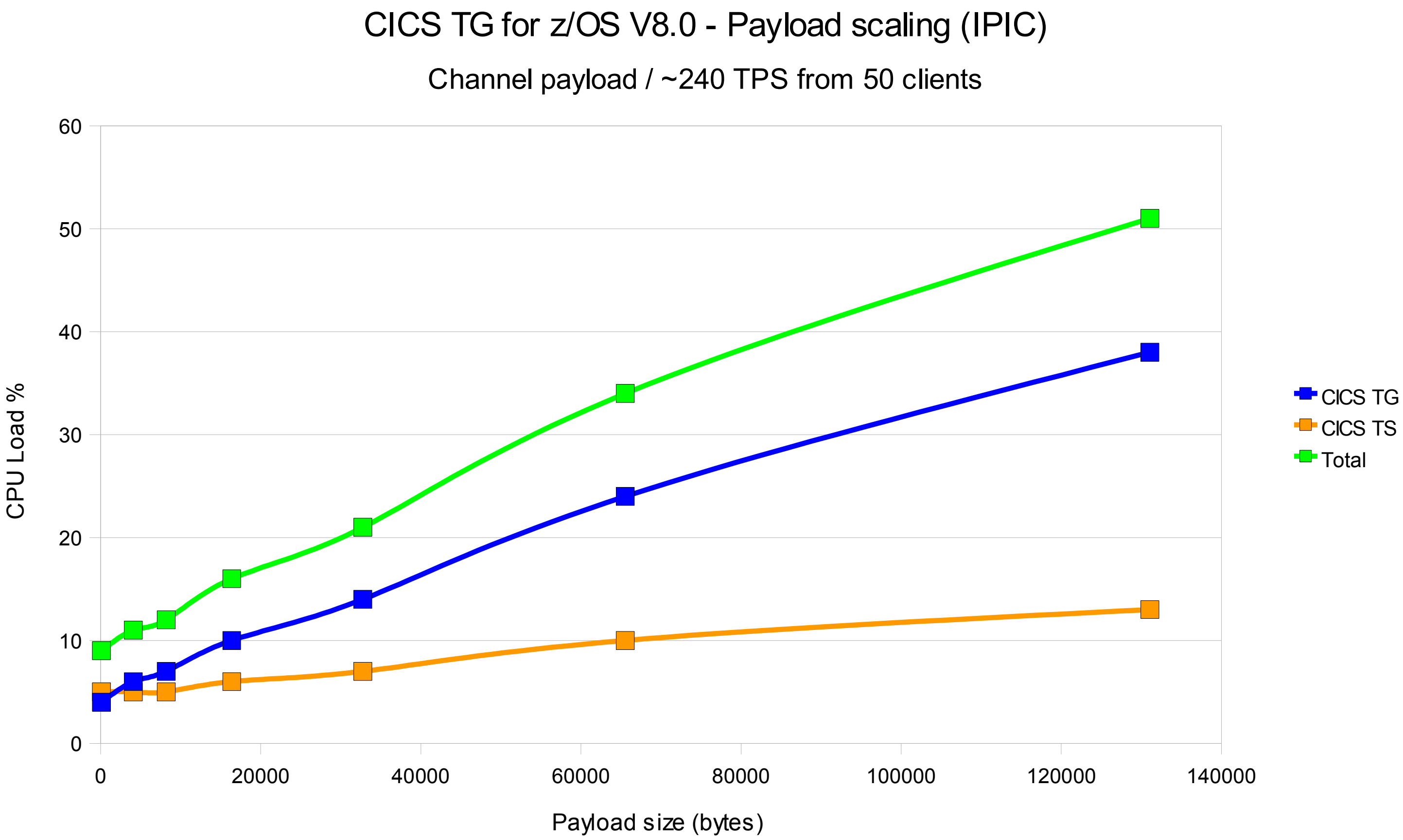

Illustration 4: Channel payload / ~240 TPS from 50 clients

Notes
In this scenario, a stable rate of requests (240 TPS) is maintained whilst the size of the CHANNEL payload sent/received by the client application increases from 100 bytes to 128KB.

This shows that IPIC connections scale well for ECI CHANNEL requests as payload sizes increase beyond 32KB. CPU usage of CICS TG and CICS TS is proportional to average payload size of the workload.

The range of transaction rates here (and hence CPU load) are lower compared with the COMMAREA scenario, since the number of clients were 50 rather than 150.

IPIC client scaling with COMMAREA

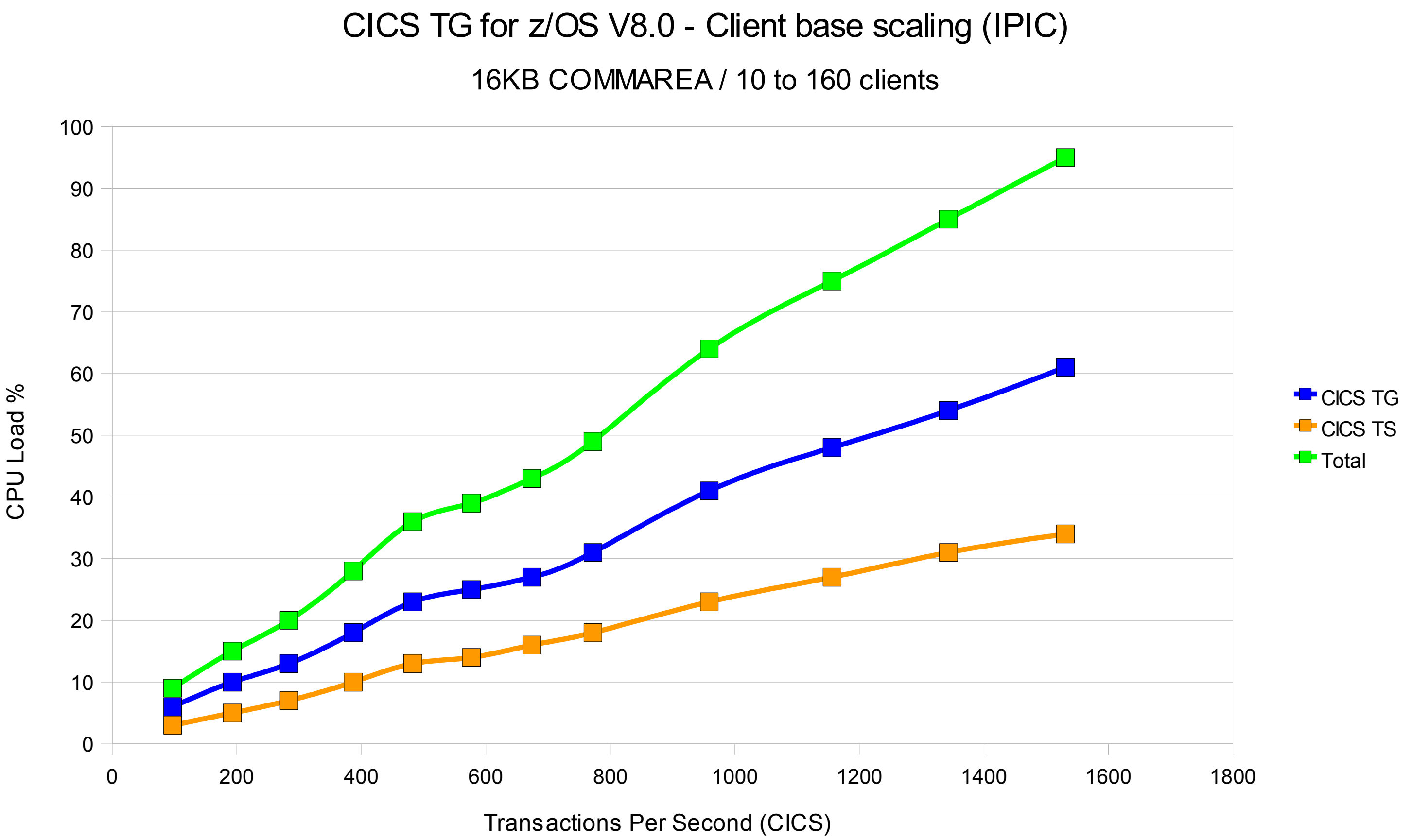

Illustration 5: 16KB COMMAREA / 10 to 160 clients

Notes

In this scenario the COMMAREA payload size (16KB) and frequency of requests from each client is maintained, whilst the number of clients increases from 10 through to 160.

This shows that IPIC connections scale well as the number of concurrent requests increases. CPU usage of CICS TG and CICS TS is proportional to the transaction rate, as the number of concurrent requests increases.