IBM CICS Transaction Server V5.1 and WebSphere MQ V7.10

A Performance Comparison of CICS using VTAM LU2 and CICS-WebSphere MQ Adapter

March 2014

IBM Hardware Acceleration Lab

Terry Borden
Greg Dyck
Nicholas C. Matsakis
Hao Wu
Xue Lian Zhang (Shelley)
Chun-Xin Yang

Overview

In this paper we compare the performance of a CICS® OTE (Open Transaction Environment – threadsafe) mode workload being driven by 3270-based terminals (VTAM LU 2) to the same workload being driven by Websphere MQ clients connected to CICS® via the CICS®-WebSphere® MQ Adapter. In both cases, non persistent requests were made. Neither recoverable persistent WebSphere MQ messages nor VTAM protected conversations were used.

We used the CICS RTW workload running DB2 V10, CICS 5.1 and z/OS® V2R1 on a IBM zEnterprise EC12 HA1 machine with 8, 16, 24, and 32 dedicated CPs. The RTW workload is one of the traditional IBM CICS®/DB2® workloads.

Apart from the number of CPUs and the workload's external delivery mechanism, there were no other configuration differences between any of the measurement runs. The WebSphere MQ Channel definitions, Channel Initiator (CHINIT) Adapter and Dispatcher setting were tuned to achieve optimum performance for the workload at the larger number of CPUs and then held constant across all the test runs.

For each CP point and front end type (VTAM/MQ), the following monitoring tools were used to measure the workload:

- RMF™ (Resource Measurement Facility) measurement data was used to compare the number of transactions per second (ETR) and the total LPAR (Logical Partition) CPU (Central Processing Unit) busy percentage
- The CICS Statistics Utility Program was used to collect CICS task busy rates
- The OMEGAMON® XE for DB2 Performance expert was used to collect DB2 and IRLM latch contention rates.

Result Highlights

This paper may be useful in understanding the potential difference between using a VTAM and WebSphere MQ delivery mechanism for CICS® based workloads. It may also be useful in understanding the benefits of different WebSphere MQ configurations.

The measurements showed the following points:

- The transaction rate per second (ETR) scaled nearly linearly through 16 CPs and started to tail off at 24 CPs. This is almost all due to DB2/IRLM internal contention and not related to MQ, CICS, or VTAM resource consumption.
- Using WebSphere MQ the workload could achieve a similar ETR with lower CPU utilization.
  - WebSphere MQ showed that it was more efficient within the CICS regions as the L8 Pattern Message Handler Transaction tasks were constantly busy getting a request from the inbound queue, processing the target transaction by linking to the target program, and putting the response on the response queue for a continuous flow of messages without having to terminate. Unlike the

VTAM runs, this resulted in very little task switching between the L8 tasks and the QR task. For VTAM, each terminal request and response is processed under the QR task and the L8 tasks are used to process each instance of the associated transaction. An environment where there is a slower flow of in bound messages such that the inbound queues go from non-empty to empty more frequently may see different results as this would driver more MQ and CICS processing to stop and restart the CICS MQ pattern transaction. How frequently this occurred would determine the additional cost.

- When using a single DB2 plan and single RTW database for all the transaction, the increase in MQ efficiency showed a higher ETR for the 8 and 16 way runs. However, at the higher 24 and 32 way, VTAM started to show a slight improvement in ETR because it did not cause as much DB2 contention.

Comparison of RTW VTAM and RTW WebSphere MQ

3270 CICS/DB2 workload (RTW)

RTW is a standard workload used by the CICS Hursley Performance team to assess changes in performance characteristics within new releases of CICS code when running DB2 applications. In these applications the presentation logic is separated from the business logic using an EXEC CICS LINK invocation of the application logic.

The workload as used in this paper has the following characteristics:

- All programs written in COBOL
- 7 unique CICS transactions that can run in any AOR
  - Manipulate 20 database tables
  - Contain SQL consisting of 54% select, 1% insert, 1% update, 1% delete, 8% open cursor, 27% fetch cursor, and 8% close cursor
- Terminal and application processing is included in the same region rather than separate TORs and AORs. The CICS MVS High Performance Option (HPO) was enabled.

The front-end presentation logic is very simple. It receives data from the terminal, passes it to the back-end business logic, and sends a response to the terminal when the logic returns control. The workload was designed to be run either quasi-reentrant or threadsafe. For this evaluation the workload was run in threadsafe mode. When the workload was run in CICS OTE mode no non-threadsafe transactions were running.

Network simulation

In the VTAM 3270 terminal configuration, the network is simulated using TPNS LU2 terminals. TPNS is run on a separate z/OS system to avoid affecting the CPU usage of the system under test. The transaction rate is changed dynamically by altering the millisecond ‘user think time’ (UTI). For this evaluation we have used different UTI values (i.e., creating different transaction arrival rates) to achieve approximately 90% processor utilization in order to compare the transaction rates across different CPU number configurations.

For the CICS-WebSphere MQ Adapter configuration, we added peer CICS regions on the TPNS system which are driven by the same TPNS network. A terminal request results in a CICS transaction which does an MQPUT to a request queue targeted to a peer region on the system under test and then it does an MQGET waiting for the response. This allowed the use of the same UTI mechanism for controlling the transaction rate in order to tune the CPU utilization to as close to 90% as possible.

In both test configurations, TCP/IP was used for communications between the systems with Enterprise Extender used to carry the SNA traffic over IP networks.

Adaptations for WebSphere MQ

To facilitate the WebSphere MQ front end, we:

1. Modified the TPNS driving system to queue requests to WebSphere MQ request queues and wait for the response on a reply queues. There was 1 request and reply queue per target CICS region. To handle the number of requests, 1 WebSphere MQ channel in each direction was used between the sending system and each CICS Region in the receiving system.

2. To manage the inbound request queue dispatching on the receiving system, the CICS-Websphere MQ Adapter Pattern was used. It is available at: http://www.ibm.com/developerworks/websphere/library/techarticles/0511_suarez/0511_suarez.html.

   The pattern consists of Cobol CICS transaction programs that manage the WebSphere MQ Inbound queue trigger processing and the dynamic management of CICS transactions required to process the messages on the inbound queue. A controller transaction monitors the inbound queue depth and when it gets too deep it initiates another handler transaction to start consuming request elements from the inbound request queue. The processing of the requests consists of an MQGET, calling the appropriate RTW transaction to handle the request, and an MQPUT of the response onto the response queue. The handler transaction does this processing in a loop. The maximum number of handlers was set to match the maximum number of OTE RTW transaction processing tasks that ran in the equivalent RTW VTAM CICS regions. More details about the Pattern Code and why it was chosen can be found at the referenced website.

   .

Environment

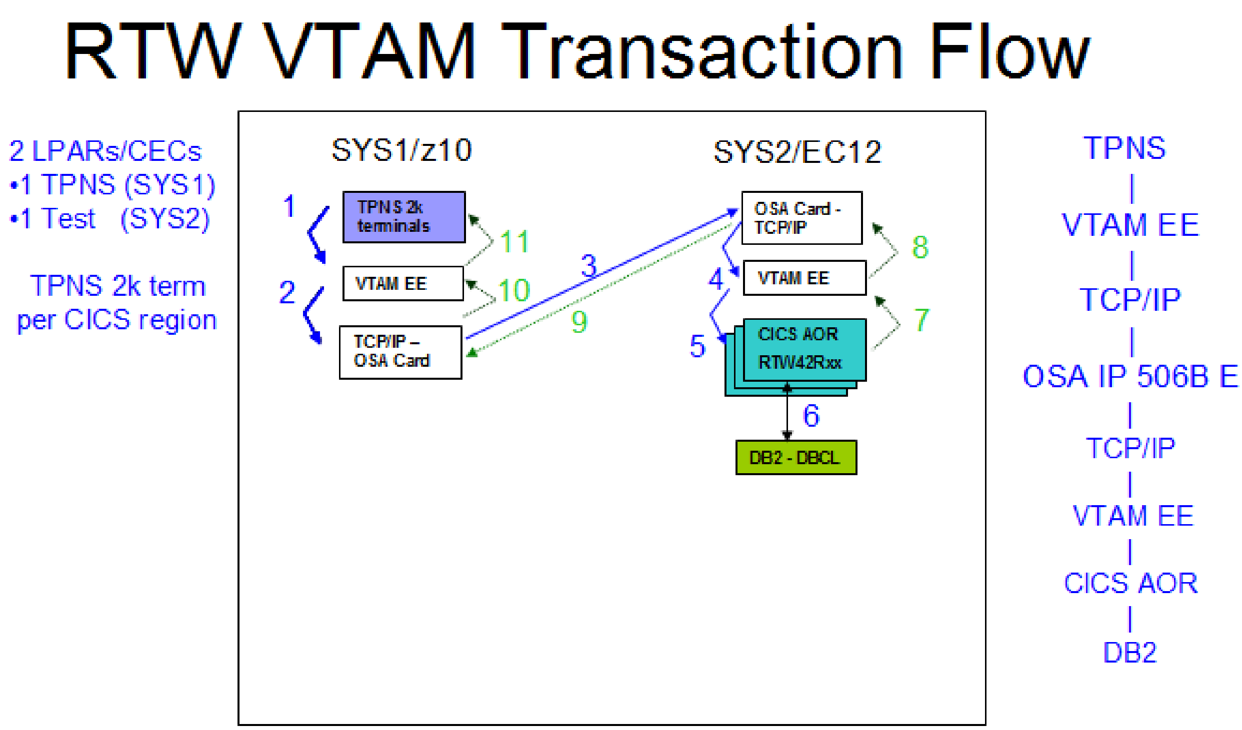
- EC12 with up to 32 dedicated CPs
- TPNS on a separate system with message sizes around 600 bytes.
- DS8800 DASD
- z/OS V2R1
- DB2 10 for z/OS
- CICS TS V5.1
- WebSphere MQ V7.1.0

Performance data collection

RMF and CICS Statistics Utility were used to measure three important metrics for this comparison:

- External Transaction Rate (ETR) which represents the number of RTW transactions that were completed per second.
  - For the VTAM environment, the ETR was reported by RMF by assigning the CICS APPLIDs to a unique WLM reporting group in the CICS subtype.
  - For the WebSphere MQ environment, WLM classification rules could not be used because the classification would have to be on a CICS transaction basis and the CICS-WebSphere MQ Adapter starts long running CICS transitions that loop doing MQGET/MQPUTs until the receive/inbound queue is empty for a given amount of time. Each element that is processed represents an individual transaction that needs to be counted. The adaptor code CICS links to the appropriate RTW transaction related “business logic” routine. As the WLM/RMF approach could not be used, WebSphere MQ queuing statistics as well as the CICS Statistics Utility Program (DFHSTUP) were used to determine the number of transactions that were processed for each test. The WebSphere MQ statistics were used to determine the total number of queue elements that were processed and they were cross referenced with the CICS statistics that indicated the number times that the transaction related “business logic” routine was called by CICS.
- The average LPAR CPU busy percentage reported in the CPU Activity Report.
- The Internal Throughput Rate (ITR) was computed by dividing the ETR by the CPU % times 100. ITR measures transaction rates per second of CPU resource efficiency. You get a higher ITR for less CPU and a lower ITR for more CPU at the same transaction rate.

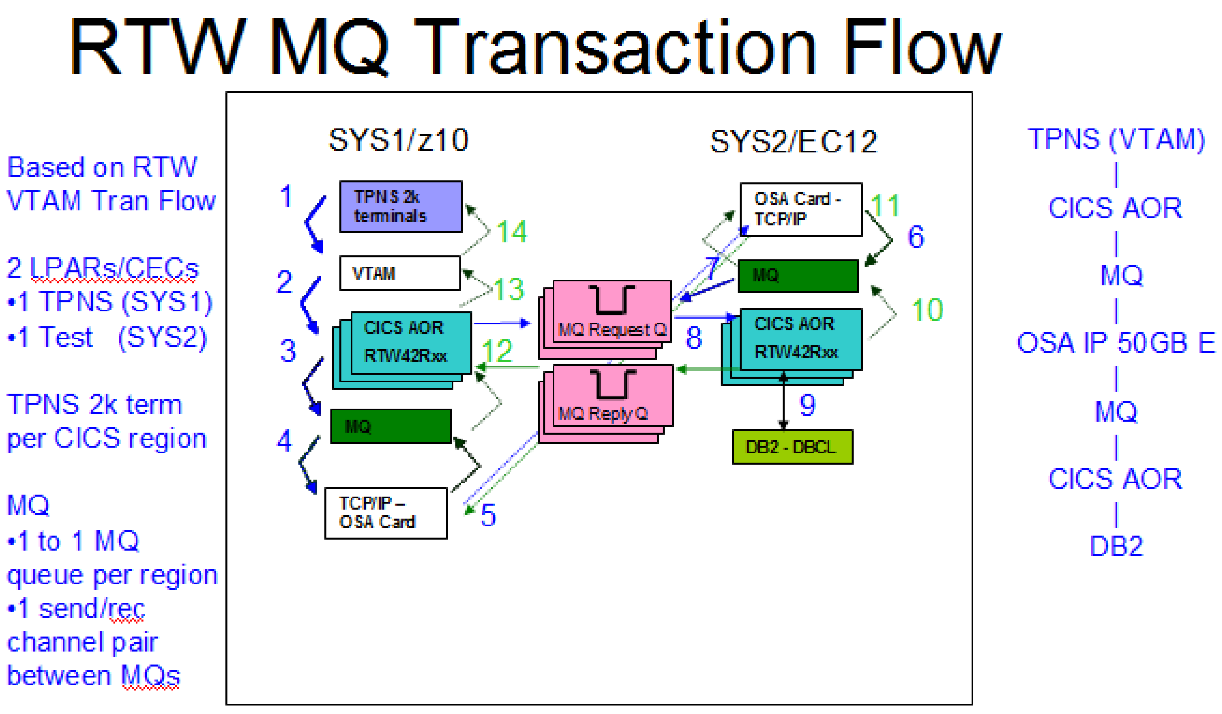
VTAM Test Workload Flow



Transactions start on the TPNS system via 2000 3270 terminals per CICS region and end with responses to those terminal requests. Data flowing to the target system under test (SYS2) are indicated by blue arrows and numbers. Green arrows and numbers show replies flowing back to the originating system (SYS1). The order is a follows:

1. SYS1: TPNS starts a transaction via a terminal request.
2. SYS1: VTAM processing sends the request to be processed by the target CICS region on SYS2
3. SYS1 VTAM EE, to TCP/IP OSA card to OSA card on SYS2
4. SYS2 TCP/IP to VTAM EE
5. SYS2 VTAM to CICS AOR handling the unique terminal
6. SYS2: The application code runs under an OTE thread whose SQL statements are handled by DB2 and the application code returns
7. through 11: The response flows back through the same communications layer to the TPNS terminal

WebSphere MQ Test Workload Flow



Like the VTAM only environment, transactions start on the TPNS system via 2000 3270 terminals per CICS region and end with responses to those terminal requests. However, the target VTAM terminal handing AORs are on the TPNS system (SYS1) which uses MQ to send requests to AORs on the system under test (SYS2).

Data flowing to the target system under test (SYS2) are indicated by blue arrows and numbers. Green arrows and numbers show replies flowing back to the originating system (SYS1). The order is a follows:

1. SYS1: TPNS starts a transaction via a terminal request.
2. SYS1: VTAM processing sends the request to be processed by the target CICS region
3. SYS1: CICS AOR picks up request and starts transaction. The transaction does an MQPUT on request queue and waits via an MQGET with sequence number from the reply queue
4. SYS1: WebSphere MQ sends the message via a TCP/IP packet (several msgs) over the OSA card to the SYS2 EC12 system
5. SYS2: TCP/IP delivers the packet
6. SYS2: WebSphere MQ puts the inbound msg on the request queue and triggers the initiation queue exit in the CICS region if needed
7. SYS2: One of the CICS-WebSphere MQ Adapter pattern message handler task's MQGETs completes and it runs the application code to handle the transaction
8. SYS2: The application code runs under an OTE thread whose SQL statements are handled by DB2 and the application code returns
9. SYS2: The CICS-WebSphere MQ Adapter code does an MQPUT with the results to the reply queue
10. SYS2: WebSphere MQ sends the response over TCP/IP to the target QMGR who is managing the reply queue.
11. SYS1: WebSphere MQ on the sending system delivers the message to the CICS transaction that was MQPUT waiting for the for the reply
12. SYS1: The CICS transaction replies to the VTAM terminal and completes
13. SYS1: TPNS terminal processing completes the transaction and starts another one.

Tuning and Configuration

- **WebSphere MQ**
  - CHINIT settings
    - 1 Channel in each direction for each CICS region under test plus an additional 20 channels. If 16 regions, then we had 36 channels defined.
    - 4 Adapter tasks
    - 8 dispatcher tasks

    Testing showed that this number of Adapters and dispatchers worked well so we kept this relationship for all of the tests.
  - QMGR
    - 1 send and receive queue per CICS region. Send Queues local to the system under test. Receive queues local to the sending TPNS system.
    - Some tests were done at 24 and 32 CPs with multiple queue managers in order to determine if internal MQ was a possible bottleneck.
- **CICS**
    - Previous testing showed that we achieved good scaling by setting the number of CICS regions to half the number of CPs and keeping the number of CICS

OTE tasks per region consistent across all measurements. It may be that varying the number of tasks and regions may not have had similar results

- CICS Adapter Pattern
  - Other than setting the limits to our liking, adding diagnostics and adding the CICS task number to the output messages, no other changes were made.

- **DB2CONN**:
  - THREADLIMIT(8) PRIORITY(EQUAL) TCBLIMIT(8) REUSELIMIT(10000)
- **DB2ENTRY**:
  - PROTECTNUM(8) THREADLIMIT(8)
- **DB2**
  - One subsystem in non-datasharing mode was used.
  - CTHREAD set to ensure that the threshold was never reached.
  - Log data sets striped over 4 dedicated volumes on separate DS8800 control units
  - Buffer pools tuned with best practice guidelines
  - Note that in order to reduce contention on IRLM and DB2 resources, 24 and 32 CP tests were done with Multiple DB2 plans, one for each AOR, and two RTW databases. More information is provided below and these are referred to as “Multiple” test runs.

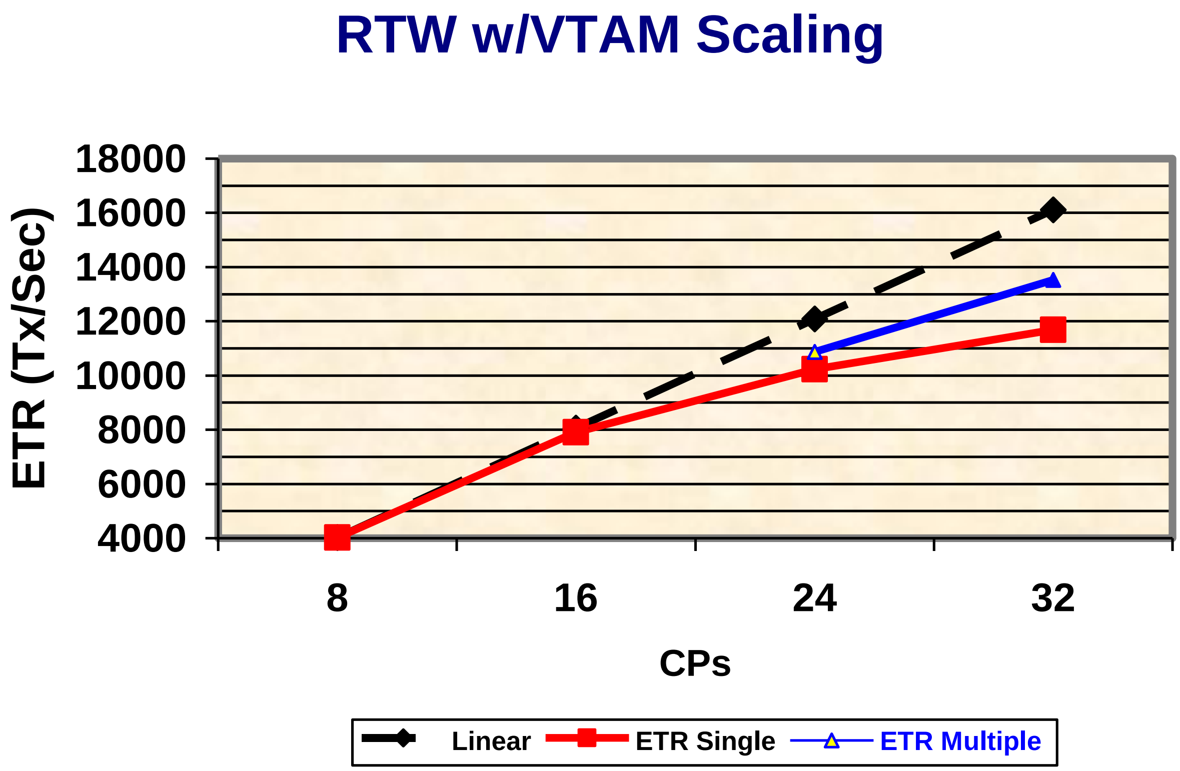
The QR and L8 TCBs were defined with an EQUAL dispatch priority. Some configurations may show a slight benefit with HIGH priority for the L8 TCB, but overall an EQUAL dispatch priority provided the best results for the RTW workload so that was held constant across all the tests.

Results

VTAM 3270-based RTW benchmark results

The following table and chart show the number of regions and terminals used for each measurement point and the resulting ETR, ITR, LPAR CPU busy percentage and overall IRLM and DB2 contention counts. All the CP test points were done with a single shared DB Plan and RTW Database. These tests showed that the ETR scaled nearly linearly through 16 CPs and started to tail off at 24 CPs. It appeared that this was due to DB2/IRLM internal contention and that it was not related to MQ, CICS, or VTAM resource consumption. As such, additional VTAM and MQ test runs were also done at 24 and 32 CP points with Multiple DB2 plans, one for each AOR, and two RTW databases. These tests are qualified with "Multiple" in the data below. Tests with a single shared plan and RTW database are qualified with "Single".

CPs	Regions	Terminals	Plans/ DBs	ETR	LPAR CPU %	ITR	IRLM Locking Susp Quantity (k)	DB2 Latch Susp Cnt/Sec	RUNID
8	4	8000	Single	4028	91.8	4387	230	1908	Z4041PBA
16	8	32000	Single	7902	91.4	8645	1203	10743	Z4042PBA
24	12	48000	Single	10230	90.9	10239	5105	34157	Z4043PBC
32	16	64000	Single	11684	87.9	13288	10293	49374	Z4052PBA
24	12	48000	Multiple	10875	91.0	11955	1476	22531	Z4059PB1
32	16	64000	Multiple	13521	91.4	14802	3990	53722	Z4059PB3



RTW w/VTAM Scaling
18000
16000
14000
12000
10000
8000
6000
4000
ETR (Tx/Sec)
8
16
24
32
CPs
Linear
ETR Single
ETR Multiple

WebSphere MQ based RTW benchmark results

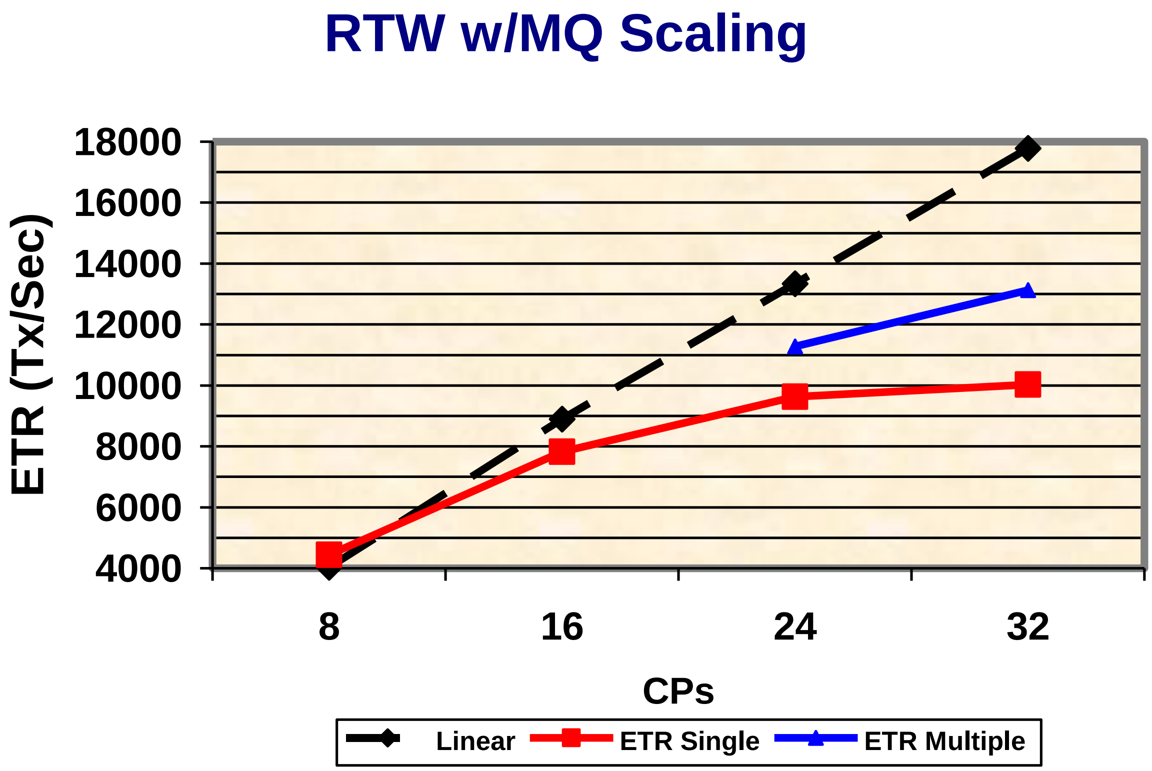
The following table shows the number of regions used for each measurement point and the resulting metrics. The subsequent graph shows the measurement results for the RTW workload running with a single WebSphere MQ V7.1.0 Queue Manager mode.

Like the VTAM runs, the ETR does not scale linearly starting at 24 CPs and using multiple DB2 plans and RTW databases resulted in better throughput due to less IRLM and DB2 resource contention. These tests are qualified with "Multiple" in the data below. Even though the "Multiple" tests showed a benefit for both VTAM and MQ, they provided more benefit for MQ.

Tests were also done with two Queue Managers to determine if MQ internal MQ contention was causing the lower ETR but they didn't show any increase in ETR at 24 or 32.

CPs	CICS Regions	Plans/ DBs	ETR	LPAR CPU %	ITR	IRLM Locking Susp Quantity (k)	DB2 Latch Susp Cnt/Sec	RUNID
8	4	Single	4446	89.0	4994	701	4608	Z4041PBC
16	8	Single	7818	84.2	9284	2774	21156	Z4043PBB
24	12	Single	9632	78.8	12337	5485	28642	Z4048PBB
32	16	Single	10029	77.1	13008	18329	67821	Z4052PBB
24	12	Multiple	11277	89.0	12579	5894	38952	Z4055PBH
32	16	Multiple	13132	87.6	14989	6978	79174	Z4059PB4

A perfectly linearly scaling workload would provide the same ETR value for each CPU that is added. The linear scaling lines in the following chats show what the perfect values would be and the actual values show how much the workload deviated from it.


RTW w/MQ Scaling
18000
16000
14000
12000
10000
8000
6000
4000
ETR (Tx/Sec)
8
16
24
32
CPs
Linear
ETR Single
ETR Multiple

Conclusion

The RTW workload using WebSphere MQ or VTAM performs similarly with VTAM achieving less ETR at 8 and 16 CPs but higher ETR at 24 and 32 CPs. However, the higher ETR is at the cost of a lower ITR for all CP points except 32. Different arrival rates to DB2 cause different DB2 contention rates which has a direct effect on ETR and ITR. Given that this workload causes a constant flow of work against the same database resources, MQ seems to run more efficiently and thus achieves a better ITR but this efficiency increases DB2 latch contention which at higher CP points reduces the ETR.

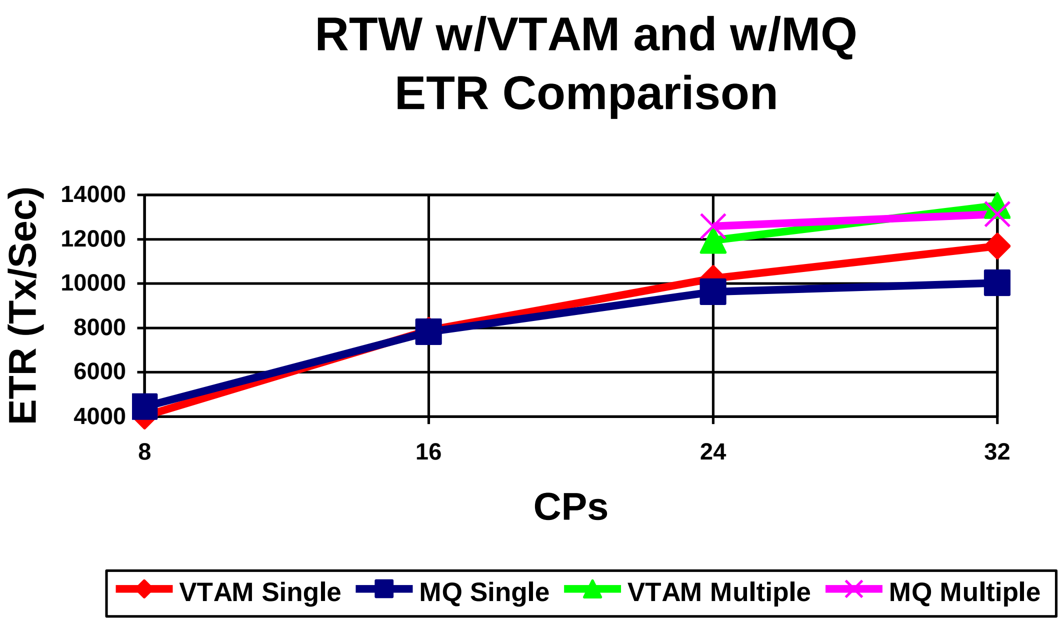
This study only used a fully loaded RTW workload and was focused specifically on scaling differences. There are many other factors such as quality of service, workload balancing, ease of use, etc. that need to be considered when deciding which communications method to use.

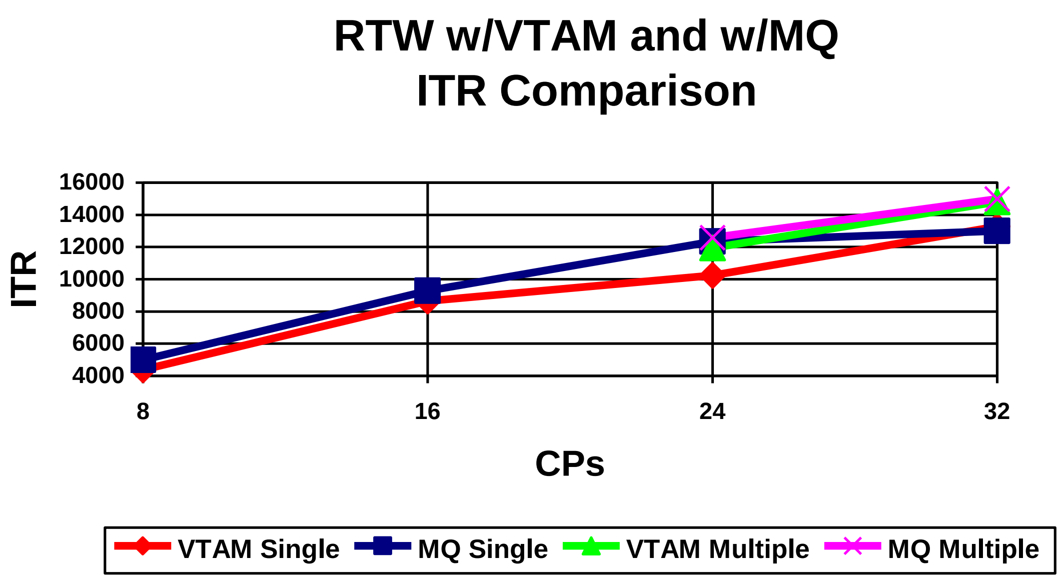
As shown by the DB2 and IRLM contention statics, they were the main bottleneck and it is believed that differences in contention were due to different arrival rates. As every customer environment varies with regard to these rates and probably few would be as high as we were measuring per CP for the same database resource, the contention related to similar RTW-like database specific resources would most likely be less. The DB2 log related resources such as the log latch, which is shared for all of DB2, may be similar at such overall rates.

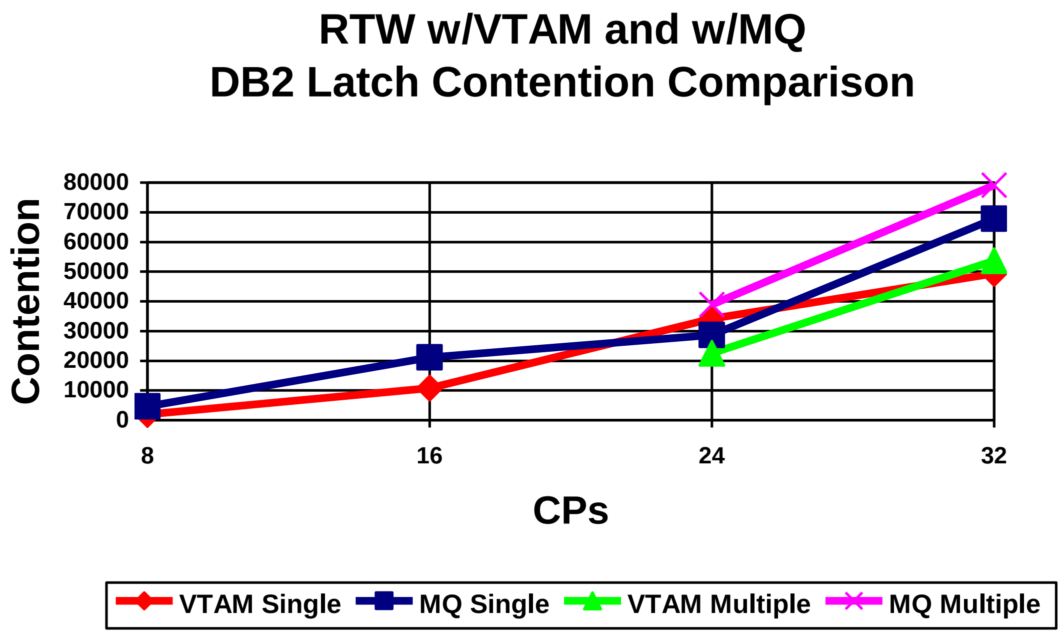
The following graphs compare the ETR, ITR, and both IRLM and DB2 contention achieved by each mode (VTAM and WebSphere MQ) as the number of CPs increased from 8 to 32. Multiple DB2 Plan and RTW Database (DB) runs were only done at 24 and 32 CPs as at least 90% CPU utilization was achieved at 8 and 16 CPs. The DB2 and IRLM contention counts were obtained from the OMEGAMON XE FOR DB2 PERFORMANCE EXPERT data reports in these sections:

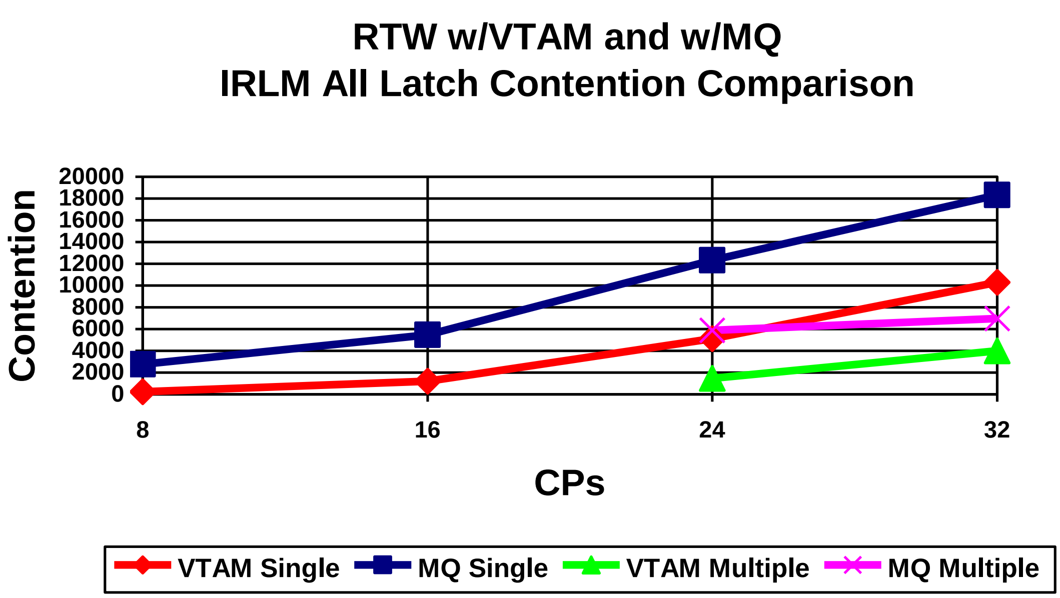
- IRLM contention was obtained from the SUSPENSIONS(ALL) QUANITY reported in the LOCKING ACTIVITY section.
- DB2 Latch Contention was obtained from the same report under the LATCH CNT section where contention counts for each latch LCxx are listed.

The ETR and ITR graphs show that when using a single plan and RTW Database, MQ provides better ETR up to 24 CPs but has better ITR for all the points except the 32w where they were about the same. The DB2 contention and IRLM contention charts show evidence that the MQ runs are causing more contention in IRLM and DB2 probably due to faster arrival rates as it is more efficient within the CICS regions.


RTW w/VTAM and w/MQ
ETR Comparison
ETR (Tx/Sec)
14000
12000
10000
8000
6000
4000
8
16
24
32
CPs
VTAM Single
MQ Single
VTAM Multiple
MQ Multiple


RTW w/VTAM and w/MQ
ITR Comparison
ITR
16000
14000
12000
10000
8000
6000
4000
8
16
24
32
CPs
VTAM Single
MQ Single
VTAM Multiple
MQ Multiple


RTW w/VTAM and w/MQ
DB2 Latch Contention Comparison
Contention
80000
70000
60000
50000
40000
30000
20000
10000
0
8
16
24
32
CPs
VTAM Single
MQ Single
VTAM Multiple
MQ Multiple


RTW w/VTAM and w/MQ
IRLM All Latch Contention Comparison
Contention
20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
8
16
24
32
CPs
VTAM Single
MQ Single
VTAM Multiple
MQ Multiple

Reference material

- IBM A pattern for implementing the CICS-WebSphere MQ Adapter
- IBM CICS TS 5.1 Application Programming Guide, SC34-7158
- IBM CICS TS 5.1 Application Programming Reference, SC34-7159